

**DIMENSIONAMENTO DE PILARES ESBELTOS DE CONCRETO
ARMADO COM SEÇÃO QUALQUER SOLICITADOS
POR FLEXÃO COMPOSTA OBLÍQUA**

ISMAEL WILSON CADAMURO JÚNIOR

Dissertação apresentada à Escola de Engenharia de São Carlos, da Universidade de São Paulo, como parte dos requisitos para obtenção do Título de Mestre em Engenharia de Estruturas.

ORIENTADORA: Prof^ª. Dra. Ana Lúcia Homce de Cresce El Debs

São Carlos
1997

Ofereço aos meus pais
esta e todas as conquistas da minha vida.
Dedico também aos meus irmãos
mais esta vitória.

AGRADECIMENTOS

À Professora Ana Lúcia Homce de Cresce El Debs, pelo trabalho de orientação, atenção, estímulo e compreensão.

À Professora Mildred Ballin Hecke, pelo incentivo e apoio.

A todos os colegas, professores e funcionários do Departamento de Estruturas da EESC/USP pela amizade.

À CAPES e à FAPESP, pelo apoio financeiro.

SUMÁRIO

Lista de Figuras	i
Lista de Abreviaturas e Siglas	iii
Lista de Símbolos	iv
Resumo	xii
<i>Abstract</i>	xiii
<u>1. INTRODUÇÃO</u>	01
1.1. Generalidades	01
1.2. Importância do assunto	02
1.3. Objetivos do trabalho	03
1.4. Revisão da literatura	03
1.5. Descrição sucinta dos capítulos	05
<u>2. HIPÓTESES DE CÁLCULO</u>	07
2.1. Manutenção da seção plana	07
2.2. Manutenção da forma da seção	07
2.3. Solidariedade dos materiais	07
2.4. Inexistência de tensões e deformações iniciais	07
2.5. Pequenos deslocamentos	08
2.6. Consideração da fluência	08
2.7. Tipo de seção transversal admitida	09
2.8. Convenção de sinais de forças, tensões, deformações, momentos fletores e de curvatura	10
<u>3. EQUACÕES GERAIS</u>	13
3.1. Equação de equilíbrio	13
3.2. Equação de compatibilidade de deformações	14
3.2.1. Equação de compatibilidade geral	14
3.2.2. Equação de compatibilidade no ELU	17
3.3. Equações constitutivas	20
3.3.1. Aço	20
3.3.2. Concreto	21

<u>4. CÁLC. DOS ESFORÇOS RESISTENTES DA SECÇÃO</u>	26
4.1. Apresentação do problema	26
4.2. Cálculo da contribuição resistente do aço	27
4.3. Cálculo da contribuição resistente do concreto	28
4.3.1. Comentários iniciais	28
4.3.2. Principais métodos de integração das tensões ao longo da seção transversal	29
4.3.3. Processo da poligonal com integração analítica	31
4.3.4. Considerações sobre a precisão	37
4.3.5. Obtenção da poligonal comprimida	39
4.3.6. Cálculo da contribuição resistente do concreto no caso de estado de deformação uniforme ($\theta = 0$)	41
<u>5. DETERMINAÇÃO DO ESTADO DE DEFORMAÇÃO DA SECÇÃO</u>	44
5.1. Introdução	44
5.2. Apresentação do problema	45
5.3. Processos iterativos para a determinação do estado de deformação da seção	46
5.4. Processo iterativo para encontrar ε_{CG} que iguale N_r a N_s	47
5.5. Processo iterativo para encontrar α que iguale β_r a β_s	52
5.6. Processo iterativo para encontrar θ que iguale $ M_r $ a $ M_s $	59
<u>6. ANÁLISE DE SEÇÕES NO ELU</u>	66
6.1. Introdução	66
6.2. Verificação da capacidade resistente de seções	66
6.2.1. Apresentação do problema	66
6.2.2. Processo iterativo para verificação de seções	69
6.2.3. Processo iterativo para encontrar ε_{CG} e θ que iguale N_r a N_s	70
6.2.4. Processo iterativo para encontrar α que iguale β_r a β_s	71

6.3. Dimensionamento da armadura de seções	74
<u>7. ANÁLISE DA ESTABILIDADE DE PILARES</u>	76
7.1. Introdução	76
7.2. Verificação da estabilidade pelo método do equilíbrio com o processo exato	76
7.2.1. Processo exato	76
7.2.2. Discretização do pilar	76
7.2.3. Cálculo dos deslocamentos transversais	79
7.2.4. Cálculo dos momentos de 2ª ordem	83
7.2.5. Processo iterativo para a verificação da estabilidade pelo método do equilíbrio com o processo exato	84
7.3. Verificação da estabilidade pelo método do equilíbrio com o processo do pilar padrão	86
7.3.1. Processo do pilar padrão	86
7.3.2. Processo iterativo para a verificação da estabilidade pelo método do equilíbrio com o processo do pilar padrão	87
7.4. Verificação da estabilidade pelo método geral	88
7.5. Dimensionamento da armadura de pilares esbeltos	90
<u>8. APLICAÇÕES NUMÉRICAS</u>	92
8.1. Exemplo 1	92
8.2. Exemplo 2	93
8.3. Exemplo 3	93
8.4. Exemplo 4	94
8.5. Exemplo 5	95
8.6. Exemplo 6	96
<u>9. CONCLUSÕES E COMENTÁRIOS</u>	98
<u>ANEXO A - MÉTODOS NUMÉRICOS PARA O CÁLCULO DE ZEROS DE FUNÇÕES</u>	100

<u>REFERÊNCIAS BIBLIOGRÁFICAS</u>	116
--	-----

<u>APÊNDICE 1 - LISTAGEM DO SISTEMA FLEXOR</u>	122
---	-----

LISTA DE FIGURAS

FIGURA 2.1 - Exemplos de poligonais	09
FIGURA 2.2 - Excentricidades da força normal	11
FIGURA 2.3 - Deformações na seção transversal - sistema de coordenadas Ouv	12
FIGURA 3.1 - Estado de deformação de uma seção transversal	16
FIGURA 3.2 - Pólos de ruína do ELU	17
FIGURA 3.3 - Diagramas tensão-deformação para os aços	21
FIGURA 3.4 - Exemplo de diagrama tensão-deformação para o concreto	22
FIGURA 3.5 - Diagrama tensão-deformação Parábola-Retângulo	23
FIGURA 3.6 - Diagramas σ - ε para cargas de curta e de longa duração	24
FIGURA 3.7 - Possíveis valores de f_c para o estudo da estabilidade	24
FIGURA 4.1 - Sistema de coordenadas Ouv	27
FIGURA 4.2 - Processo das Malhas - tipos de elementos	30
FIGURA 4.3 - Decomposição da poligonal em trapézios	31
FIGURA 4.4 - Integração das tensões em um trapézio genérico	32
FIGURA 4.5 - Exemplo de poligonais origem e resultado	39
FIGURA 5.1 - Esquema geral do processo iterativo para a determinação do estado de deformação da seção	47
FIGURA 5.2 - Seção transversal teste	49
FIGURA 5.3 - Andamento típico da função $N_r=f(\varepsilon_{CG})$	49
FIGURA 5.4 - Andamento típico da função $\beta_r=f(\alpha)$	53
FIGURA 5.5 - Andamento típico da curva $M_{rx}-M_{ry}$	53
FIGURA 5.6 - Andamento da função $\beta_r=f(\alpha)$ para seção assimétrica e θ baixo	55
FIGURA 5.7 - Andamento da curva $M_{rx}-M_{ry}$ para seção assimétrica e θ baixo	55
FIGURA 5.8 - Andamento típico do diagrama M-N- θ para seção simétrica	60
FIGURA 5.9 - Andamentos típicos do diagrama M-N- θ para seção assimétrica	61
FIGURA 5.10 - Processo iterativo x ELU	62
FIGURA 5.11 - Exemplos de diagrama M-N- θ	65
FIGURA 6.1 - Andamentos típicos do diagrama de interação no ELU	68
FIGURA 6.2 - Esquema geral do processo iterativo para a verificação de seções	69

FIGURA 7.1 - Discretização do pilar em seções transversais	78
FIGURA 7.2 - Discretização de pilar com variação brusca de seção ou de esforço	78
FIGURA 7.3 - Vinculação admitidas para os pilares	79
FIGURA 7.4 - Convenção para discretização do pilar	79
FIGURA 7.5 - Curvaturas, rotações e deslocamentos do eixo do pilar	80
FIGURA 7.6 - Pilar deslocado	84
FIGURA 7.7 - Pilar padrão	88
FIGURA 7.8 - Diagrama carga–deslocamento	90
FIGURA 8.1 - Seção dos exemplos 1, 2 e 3	93
FIGURA 8.2 - Diagrama de interação do exemplo 3	95
FIGURA 8.3 - Pilar do exemplo 4	95
FIGURA 8.4 - Pilar do exemplo 5	96
FIGURA A.1 - Zeros reais	102
FIGURA A.2 - Intervalo [a,b] com raiz	103
FIGURA A.3 - Método dos Intervalos Encaixantes	104
FIGURA A.4 - Intervalo [a,b] que atende à precisão ϵ	106
FIGURA A.5 - Exemplo onde se tem $ f(x_{\text{raiz}}) < \epsilon$ mas $ x_{\text{raiz}} - \xi \gg \epsilon$	107
FIGURA A.6 - Exemplo onde se tem $ x_{\text{raiz}} - \xi < \epsilon$ mas $ f(x_{\text{raiz}}) \gg \epsilon$	107
FIGURA A.7 - Exemplo onde se tem $ f(x_{\text{raiz}}) < \epsilon$ e $ x_{\text{raiz}} - \xi < \epsilon$	107
FIGURA A.8 - Exemplo do método da Bissecção	108
FIGURA A.9 - x no método da Posição Falsa	110
FIGURA A.10 - Exemplo do método de Newton–Raphson	112
FIGURA A.11 - Exemplo do método da Secante	114

LISTA DE ABREVIATURAS E SIGLAS

ABNT - Associação Brasileira de Normas Técnicas

CEB - Comité Européen du Béton

CG - Centro de Gravidade da Seção Bruta de Concreto

ELU - Estado Limite Último de ruptura do concreto comprimido ou de deformação plástica excessiva da armadura tracionada

eq - Equação

FCN - Flexão Composta Normal

FCO - Flexão Composta Oblíqua

fig - Figura

LN - Linha Neutra

NB - Norma Brasileira

NBR - Norma Brasileira Registrada

NLF - Não-Linearidade Física

NLG - Não-Linearidade Geométrica

LISTA DE SÍMBOLOS

Letras Gregas Maiúsculas

Δ_{ANG} - ângulo entre um lado da poligonal e a direção da linha neutra

ΔM_{ruc} e ΔM_{rvc} - parcelas de momento fletor resistente nas direções u e v fornecidas por um lado da poligonal de concreto

ΔN_{rc} - parcela de força normal resistente fornecida por um lado da poligonal de concreto

Letras Gregas Minúsculas

α - ângulo entre o eixo x e a direção da linha neutra, medido no sentido anti-horário; parâmetro de instabilidade

α_1 - fração da força normal que produz fluência; direção da linha neutra que propicia calcular o momento fletor resistente mínimo a que uma seção pode suportar

α_2 - fração do momento fletor de primeira ordem que produz fluência; direção da linha neutra que propicia calcular o momento fletor resistente máximo a que uma seção pode suportar

β - direção do momento fletor ($\beta = \arctan\left(\frac{M_y}{M_x}\right)$)

β_r - direção do momento fletor resistente

β_{r1} e β_{r2} - direções dos momentos fletores resistentes mínimo e máximo a que uma seção pode suportar

β_s - direção do momento fletor solicitante

β_v - profundidade adimensional da linha neutra ($\beta_v = \frac{v_{LN}}{h_\alpha}$)

γ_c - coeficiente de minoração da resistência do concreto

γ_s - coeficiente de minoração da resistência do aço

ε - deformação (em ‰ e com sinal) de um ponto qualquer da seção transversal

ε_{cf} - deformação por fluência do concreto

ε_{CG} - deformação (em ‰ e com sinal) da origem do sistema de coordenadas (onde se supõe estar o CG da seção; isto, porém, não é obrigatório)

$\varepsilon_{CG,máximo}$ e $\varepsilon_{CG,mínimo}$ - valores de ε_{CG} que colocam a seção no ELU de ruptura

$\varepsilon_{CG,1}$, $\varepsilon_{CG,2}$ e $\varepsilon_{CG,3}$ - deformações na origem do sistema de coordenadas calculadas supondo-se que o estado de deformação passe pelos pólos 1, 2 ou 3

$\varepsilon_{c,total}$ - deformação total do concreto

ε_{c0} - deformação imediata do concreto

ε_s - deformação (em ‰ e com sinal) da barra de aço

$|\varepsilon_s|$ - módulo da deformação da barra de aço

ε_{yd} - deformação de escoamento do aço

ε_1 e ε_2 - deformações (em ‰ e com sinal) em 2 pontos quaisquer da seção transversal

θ - curvatura majorada adimensional ($\theta = 1000h_\alpha \frac{1}{r_\alpha}$)

θ_1 , θ_2 e θ_3 - curvaturas majoradas adimensionais calculadas supondo-se que o estado de deformação passe pelos pólos 1, 2 ou 3

μ - momento fletor reduzido ($\mu = \frac{M}{A_c \times f_c \times h}$)

ν - força normal reduzida ($\nu = \frac{N}{A_c \times f_c}$)

ξ - sistema de coordenadas auxiliar

ξ_i - coordenada ξ na seção i do pilar

σ - tensão em um ponto qualquer da seção transversal

σ_c - tensão no concreto em um ponto da seção transversal

σ_s - tensão na barra de aço (com sinal)

$\sigma_{s,i}$ - tensão na barra de aço i

ϕ - função de fluência

ϕ_{ef} - função efetiva de fluência

φ - rotação

φ_i - rotação da seção i do pilar

φ'_i - valor dado pela eq. 7.11

ϕ_1 - rotação da seção 1 do pilar

ω - taxa mecânica de armadura ($\omega = \frac{A_s \times f_y}{A_c \times f_c}$)

ω_{\min} - menor taxa mecânica de armadura que propicia a que a seção resista aos esforços solicitantes dados sem romper

Letras Romanas Maiúsculas

A e B - constantes da equação de uma reta

A_c - área da poligonal que define a seção de concreto

A_{cc} - área comprimida de concreto

A_{s_i} - área de cada barras de aço

$A_{s,i}$ - área da barra de aço i

C_k - k'ésimo coeficiente do polinômio; $k \rightarrow 0$ a m ; os coeficientes do polinômio devem representar a curva como se todo o carregamento fosse de curta duração (sem o efeito da fluência)

E_s - módulo de deformação longitudinal do aço (usualmente, $E_s=210000\text{Mpa}$)

F_n - valor dado pela eq. 4.16

M - momento fletor; constante em um método numérico iterativo de cálculo de raízes de funções

$|M|$ - módulo do momento fletor ($|M| = \sqrt{M_x^2 + M_y^2}$)

MATRIZ_C - matriz ($N_m \times$ maior m) que contém os coeficientes C_k dos N_m polinômios; a linha i desta matriz contém os coeficientes do polinômio i; estes coeficientes não devem incluir o efeito da fluência

MATRIZ_e_LIMITES - matriz ($N_m \times 2$) que contém as deformações (em ‰ e com sinal) onde inicia e termina cada uma das N_m curvas polinomiais; a linha i desta matriz contém as deformações onde inicia e termina a atuação do polinômio i; estas deformações não devem incluir o efeito da fluência

$|M_r|$ - módulo do momento fletor resistente

M_{ru} e M_{rv} - momentos fletores resistentes nas direções u e v
 M_{ruc} e M_{rvc} - momentos fletores resistentes nas direções u e v fornecido pelo concreto
 M_{rus} e M_{rvs} - momentos fletores resistentes nas direções u e v fornecido pelo aço
 $M_{rus,i}$ e $M_{rvs,i}$ - momentos fletores resistentes nas direções u e v fornecido pela barra de aço i
 M_{rx} e M_{ry} - momentos fletores resistentes nas direções x e y
 M_{rxc} e M_{ryc} - momentos fletores resistentes nas direções x e y fornecido pelo concreto
 M_{rxs} e M_{rys} - momentos fletores resistentes nas direções x e y fornecido pelo aço
 $M_{rxs,i}$ e $M_{rys,i}$ - momentos fletores resistentes nas direções x e y fornecido pela barra de aço i
 $|M_s|$ - módulo do momento fletor solicitante
 M_{sx} e M_{sy} - momentos fletores solicitantes nas direções x e y
 $M_{sx,i}$ e $M_{sy,i}$ - momentos fletores solicitantes na seção i na iteração em cálculo
 $M_{sx,i,referência}$ e $M_{sy,i,referência}$ - momentos fletores solicitantes de referência na seção i (por exemplo, os esforços solicitantes de serviço)
 M_x e M_y - momentos fletores nas direções x e y
 M_1 - momento fletor que solicita a seção 1, na direção em que se está analisando o pilar (x ou y)
 $|M_{r1}|$ e $|M_{r2}|$ - módulos dos momentos fletores resistentes mínimo e máximo a que uma seção pode suportar
 M_2 - momento de 2ª ordem
 M_{2i} - momento fletor de segunda ordem que solicita a seção i, na direção em que se está analisando o pilar (x ou y)
 M_{2N_secao} - momento fletor de segunda ordem que solicita a seção N_seção, na direção em que se está analisando o pilar (x ou y)
 M_{2x} - momento de 2ª ordem na direção x
 M_{2y} - momento de 2ª ordem na direção y
 N - força normal; número de nós da poligonal que define a seção de concreto
 $N_{BARRAS DE AÇO}$ - número de barras de aço da seção transversal

$N_{contrav,i}$ - caso o pilar em análise faça parte da estrutura de contraventamento, $N_{contrav,i}$ é a soma das forças normais solicitantes, no nível i, dos pilares da estrutura contraventada

pelo pilar em análise; caso o pilar em análise não faça parte da estrutura de contraventamento, $N_{\text{contrav},i} = 0$

N_m - número de polinômios que compõem o diagrama tensão-deformação adotado para o concreto

N_r - força normal resistente

N_{rc} - força normal resistente fornecida pelo concreto

$N_{r,\text{MAXIMO}}$ - máxima força normal a que a seção transversal pode resistir

$N_{r,\text{MINIMO}}$ - mínima (ou máxima negativa) força normal a que a seção transversal pode resistir

N_{rs} - força normal resistente fornecida pelo aço

$N_{rs,i}$ - força normal resistente fornecida pela barra de aço i

N_s - força normal solicitante

$N_{\text{seção}}$ - número de seções transversais utilizadas para discretizar o pilar

$N_{s,i}$ - força normal solicitante que atua na seção i do pilar em análise

$N_{s,i,\text{referência}}$ - força normal solicitante de referência na seção i (por exemplo, os esforços solicitantes de serviço)

$N_{\text{tot},i}$ - força normal total que solicita a seção i

Ouv - sistema de coordenadas utilizado para definir as seções transversais do pilar, onde o eixo u é paralelo à linha neutra, e o eixo v é perpendicular

Oxy - sistema de coordenadas utilizado para definir as seções transversais do pilar

S_{uu} - momento estático da seção em relação ao eixo u

S_{vv} - momento estático da seção em relação ao eixo v

U_{cg} e V_{cg} - coordenadas (sist. Ouv) do CG da seção de concreto; caso tenha sido obedecida a orientação de colocar a origem do sistema de coordenadas no CG da seção, tem-se $U_{cg}=V_{cg}=0$

U_{corte} - coordenada u (sistema Ouv) em que um lado da poligonal origem é cortado para formar a poligonal resultado

$V_{\text{corte}1_i}$ e $V_{\text{corte}2_i}$ - ordenada (sist. Ouv) onde começa e termina de agir a curva i do diagrama σ - ϵ adotado para o concreto

VETOR_m - vetor ($N_m \times 1$) que contém os graus m dos N_m polinômios

Letras Romanas Minúsculas

a e b - valores dados pelas eq. 4.11 e 4.12, ou 7.5; valores do intervalo [a,b] (abscissas) em um método numérico iterativo de cálculo de raízes de funções

c - valor dado pela eq. 7.7

d - valor dado pela eq. 7.13

$|e|$ - excentricidade da força normal ($|e| = \sqrt{e_x^2 + e_y^2}$)

e_x - excentricidade da força normal ao longo do eixo x

e_y - excentricidade da força normal ao longo do eixo y

fc - resistência do concreto à compressão; usualmente, faz-se $f_c = f_{cd} = 0,85f_{ck}/\gamma_c$, com $\gamma_c = 1,4$; porém, no estudo da estabilidade, pode-se assumir $\gamma_c = 1,2$

fy - tensão de escoamento da armadura; usualmente, faz-se $f_y = f_{yd} = f_{yk}/\gamma_s$, com $\gamma_s = 1,15$; porém, no estudo da estabilidade, pode-se assumir $\gamma_s = 1,0$

h_α - altura da seção transversal, medida na direção perpendicular à da linha neutra

k - parâmetro que representa os esforços solicitantes do pilar na iteração em cálculo

k_{mola} - constante de mola da base elástica (momento necessário para causar um giro unitário), na direção em que se está analisando o pilar (x ou y)

l_e - comprimento de flambagem

l_i - lado i de uma poligonal

m - grau do polinômio (reta $\rightarrow m=0$; parábola do 2º grau $\rightarrow m=2$, etc.)

$\frac{1}{r}$ - curvatura

$\frac{1}{r_i}$ - curvatura da seção i

$\frac{1}{r_x}$ - curvatura na direção perpendicular ao eixo x

$\frac{1}{r_y}$ - curvatura na direção perpendicular ao eixo y

$\frac{1}{r_\alpha}$ - curvatura na direção perpendicular à da linha neutra

u_i e v_i - coordenadas (sist. Ouv) do nó i da poligonal de concreto

$u_{origem(i)}$ e $v_{origem(i)}$ - coordenadas (sist. Ouv) do nó i da poligonal de concreto (origem)

$u_{resultado(i)}$ e $v_{resultado(i)}$ - coordenadas (sist. Ouv) do nó i da poligonal extraída da poligonal origem

$u_{s,i}$ e $v_{s,i}$ - coordenadas (sist. Ouv) da barra de aço i

$v_{barras,min}$ - estando a seção transversal referida no sistema Ouv, $v_{barras,min}$ representa a menor ordenada das barras de aço

$v_{conc,max}$ - estando a seção transversal referida no sistema Ouv, $v_{conc,max}$ representa a maior ordenada da poligonal que define a seção transversal de concreto

v_{LN} - ordenada v (sistema Ouv) da linha neutra

v_1 e v_2 - ordenadas (sistema Ouv) dos pontos onde se medem ε_1 e ε_2

w - deslocamento transversal do eixo do pilar

w_i - deslocamento transversal da seção i do pilar

w'_i - valor dado pela eq. 7.16

w_{max} - flecha máxima do pilar padrão

$w_{max,x}$ - flecha máxima do pilar padrão na direção x

$w_{max,y}$ - flecha máxima do pilar padrão na direção y

w_x - deslocamento transversal na direção x

w_y - deslocamento transversal na direção y

w_1 - deslocamento transversal da seção 1

x - coordenada x ; valor da abscissa em um método numérico iterativo de cálculo de raízes de funções

x_a , x_b , x_0 , x_1 e x_2 - valores de abscissas em um método numérico iterativo de cálculo de raízes de funções

$x_{s,i}$ e $y_{s,i}$ - coordenadas (sist. Oxy) da barra de aço i

y_a e y_b - valores dos extremos do intervalo $[a,b]$ (ordenadas) da função em um método numérico iterativo de cálculo de raízes de funções

y_x, y_0, y_1 e y_2 - valores de ordenadas em um método numérico iterativo de cálculo de raízes de funções

z - eixo de coordenadas que contém o eixo do pilar

z_i - coordenada z na seção i do pilar

RESUMO

Este trabalho apresenta um estudo geral sobre o dimensionamento e verificação de pilares esbeltos de concreto armado solicitados por flexão composta oblíqua, levando-se em consideração as não-linearidades física (o material não segue a lei de Hooke) e geométrica (análise em 2ª ordem). É abordado também o dimensionamento de seções isoladas no Estado Limite Último de ruptura. É admitida seção transversal de forma poligonal qualquer, assim como disposição da armadura também arbitrária. O efeito de carregamento de longa duração (fluência) é levado em consideração. A relação tensão-deformação do concreto é geral, admitida como uma seqüência de polinômios de graus arbitrários. Vários algoritmos detalhados e rotinas computacionais são apresentados.

ABSTRACT

This work presents a general study on the design and verification of slender reinforced concrete columns, subjected to axial force and biaxial bending, considering material and geometric non-linearities (material doesn't fit Hooke law and second order analysis). It is also treated the design of cross sections in the ultimate strength. It is considered a general polygonal cross section, with any reinforcement distribution. The creep effect is also taken into account. The concrete stress-strain curve is anyone since it can be represented as a sequence of any degree polinomiuns. Detailed calculation and computational routines are given too.

CAPÍTULO 1 - INTRODUÇÃO

1.1. GENERALIDADES

O comportamento estrutural de um pilar esbelto de concreto armado, em geral bastante complexo, é um comportamento tipicamente não-linear. A não-linearidade física, decorrente das equações constitutivas não lineares do concreto e do aço, e a não-linearidade geométrica, caracterizada pela substancial alteração sofrida pelas solicitações em função dos deslocamentos transversais do eixo do pilar, impõem a necessidade de uma análise numérica detalhada desse elemento estrutural. O presente trabalho trata dessa análise, dedicada à pilares de seção transversal poligonal qualquer (inclusive com seção e armadura variável ao longo do comprimento do pilar) solicitados por flexão composta oblíqua (FCO). Conseqüentemente, são resolvidos todos os casos particulares do proposto (flexão composta normal (FCN), flexão simples, cálculo de flecha em vigas, etc.).

A flexão composta oblíqua ocorre sempre que, com a aplicação conjunta de força normal e momento fletor, a direção do plano do momento fletor solicitante não for coincidente com um eixo de simetria da seção (simetria tanto da seção de concreto quanto da disposição da armadura). Logicamente, se a seção não possuir nenhum eixo de simetria, a flexão será sempre oblíqua.

O estudo proposto envolve tanto a análise de seções isoladas (considerando a obediência ao ELU¹) quanto a análise da estabilidade de pilares como um todo (análise não-linear física e geométrica). Salienta-se, desde já, que o segundo problema diz respeito a pilares isolados, onde se conhecem *a priori* os esforços solicitantes de 1ª ordem. Segundo a ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS (1994), são considerados elementos isolados:

- 1) as peças isostáticas;
- 2) os elementos das estruturas contraventadas;

¹ ELU significa, neste trabalho, Estado Limite Último de ruptura do concreto comprimido ou de deformação plástica excessiva da armadura tracionada.

- 3) os elementos das estruturas de contraventamento de nós fixos (por exemplo, $\alpha \leq \alpha_{LIM}$ ou $\gamma_Z \leq 1,1 \rightarrow$ parâmetros de instabilidade);
- 4) os elementos das estruturas de contraventamento de nós móveis ($\alpha > \alpha_{LIM}$ ou $\gamma_Z > 1,1$) desde que, aos esforços nas extremidades obtidos em uma análise de 1ª ordem, sejam acrescentados os determinados por análise global de 2ª ordem (processos P- Δ , por exemplo).

Tendo em vista as peculiaridades do problema (não-linearidades física e geométrica), a solução envolve um processo iterativo e incremental complexo, tornando praticamente obrigatório o uso de computador.

O efeito da fluência é levado em consideração através da Teoria Linear de Fluência (ϕ - função de fluência).

Salienta-se que neste trabalho não é abordado o problema dinâmico (vibrações, choques, etc.) ou o de carregamento cíclico (fadiga). Também não são objeto de estudo os métodos simplificados de tratamento da FCO e da estabilidade (linearização do diagrama de interação, transformação da FCO em 1 ou 2 FCN, etc.); também não são estudados os métodos simplificados de tratamento dos efeitos da fluência (método da excentricidade equivalente, etc.).

1.2. IMPORTÂNCIA DO ASSUNTO

O assunto do presente trabalho tem grande importância, demonstrada, principalmente, pelos itens a seguir:

- a) ECONOMIA: Os métodos aproximados facultados nas normas, são, na grande maioria das vezes, muito a favor da segurança, e, portanto, contra a economia;
- b) SEGURANÇA: Alguns métodos aproximados são ora a favor, ora contra a segurança, como observado por SANTOS (1981) e SAUAIA (1981). O estudo do assunto do presente trabalho possibilita o cálculo de estruturas mais ousadas, com maior conhecimento da sua segurança (precisão/confiabilidade);

- c) POSSIBILIDADE DE CÁLCULO DE QUALQUER SEÇÃO TRANSVERSAL: É importante possibilitar ao engenheiro escolher a seção que melhor resolve determinado problema, sem ter de se restringir às seções com métodos aproximados conhecidos, principalmente face às crescentes exigências arquitetônicas² e estruturais³;
- d) POSSIBILIDADE DE CÁLCULO DE PILARES DE GRANDE ESBELTEZ: É importante possibilitar ao engenheiro o cálculo de pilares de grande esbeltez, pois as estruturas empregadas vem se tornando mais e mais esbeltas, à medida que os materiais de maior resistência se tornam mais comuns no mercado;
- e) REDUÇÃO DO TEMPO DE PROJETO: O presente trabalho possibilita a automatização do dimensionamento de pilares, reduzindo o tempo de projeto e/ou liberando mais tempo para a concepção estrutural.

1.3. OBJETIVOS DO TRABALHO

Resumidamente, tem-se como objetivos:

- a) Síntese do conhecimento sobre pilares esbeltos de seção qualquer sob FCO;
- b) Elaboração de algoritmos que funcionem com qualquer seção transversal e sob qualquer condição de carregamento;
- c) Desenvolvimento de programa computacional de cálculo;
- d) Experimentação numérica;

² como, por exemplo, o embutimento dos pilares nas paredes, impelindo ao uso de seções L, T, etc.

³ como, por exemplo, elementos de contraventamento do tipo núcleo (aproveitando caixas de elevadores, de escadas, etc.).

1.4. REVISÃO DA LITERATURA

A análise de pilares solicitados por FCO é um assunto muito estudado. Através dos anos, esse assunto foi alvo de muitas teorias e programas computacionais visando à determinação do comportamento estrutural dessas peças. A preocupação com o problema tornou-se maior quando a interação entre os pilares e as vigas do edifício passou a ser considerada no projeto, fazendo surgir uma série de estudos sobre o assunto, destacando-se BRESLER (1960) e AAS-JAKOBSEN (1964).

BUCHAIN (1979) aborda a estabilidade de pilares solicitados por FCN, utilizando o Processo do Pilar Padrão. PAULA (1988) segue a mesma linha, mas abordando também o Método Geral.

Em VENTURINI & BORTOLIN (1992), é apresentado um grande conjunto de ábacos de grande utilidade prática, dedicados ao dimensionamento de seções retangulares (seções isoladas no ELU) solicitadas por FCO. O mesmo ocorre em PINHEIRO et al. (1994).

FORNI (1980), SANTOS (1981) e SANTOS (1994) abordam, de maneira muito ampla, o estudo de seção qualquer (no ELU) solicitada por FCO. Tais estudos não levam em conta o efeito da fluência.

AUFIERO (1977) aborda o estudo da estabilidade de pilares solicitados por FCN e FCO, utilizando o Processo do Pilar Padrão. Não é abordada a seção qualquer. Nesta mesma linha, segue MARCOTTI (1984), mas abordando também o Método Geral.

FRANÇA (1984) faz um estudo detalhado das relações momento-curvatura em seções solicitadas por FCO. É admitida seção poligonal qualquer.

MENDES NETO (1991) aborda, de maneira ampla, o estudo da estabilidade de pilares de seção qualquer solicitados por FCO. O Processo do Pilar Padrão é o utilizado. CAMPOS (1994) segue nesta mesma linha, porém abordando também o Método Geral. Tais estudos não levam em conta o efeito da fluência.

Mais recentemente, artigos em revistas especializadas tratando do mesmo problema demonstram que o assunto ainda não está esgotado: AHMAD & WEERAKOON (1995),

ZIELINSKI *et al.* (1995), HOROWITZ (1995), HWEI & KEONG (1995), THEWALT & STOJADINOVIC (1994), EL-METWALLY (1994), ZAK (1993) e YAU *et al.* (1993). Basicamente, observou-se que muitos algoritmos, supostos gerais, utilizados na grande maioria dos trabalhos anteriores, apresentam problemas de convergência com seções assimétricas sob determinadas condições de carregamento (YAU *et al.* (1993)). O presente trabalho desenvolve algoritmos detalhados que são eficazes com qualquer tipo de seção transversal e sob quaisquer condições.

1.5. DESCRIÇÃO SUCINTA DOS CAPÍTULOS DO TRABALHO

Inicialmente, no capítulo *introdução*, o assunto do trabalho é apresentado e discutido; expõe-se, brevemente, os objetivos e a importância do trabalho, bem como é feita uma breve revisão da literatura existente.

No capítulo *hipóteses de cálculo*, expõe-se as hipóteses básicas iniciais, o tipo de seção transversal admitido e as convenções de sinais e de momentos fletores adotadas.

No capítulo *equações gerais*, é deduzida a formulação do problema. São montadas as equações de equilíbrio, de compatibilidade e as equações constitutivas dos materiais.

Em *esforços resistentes* da seção, são discutidos os métodos existentes de cálculo dos esforços resistentes para uma seção qualquer. Para o método escolhido, são feitas todas as deduções e é montado o algoritmo de cálculo. É dada ênfase ao cálculo da contribuição resistente do concreto (que é um dos principais problemas a ser vencido).

No capítulo *determinação do estado de deformação da seção*, é discutido minuciosamente o algoritmo que encontra o estado de deformação de uma seção, dados os esforços solicitantes e a seção definida geométrica e mecanicamente⁴. É mostrado porque os algoritmos convencionais podem não funcionar para determinados problemas. O conteúdo deste capítulo pode ser utilizado tanto na análise de pilares isostáticos quanto em análises não-lineares mais avançadas (pórticos tridimensionais hiperestáticos, por exemplo). Este é um dos capítulos mais importantes deste trabalho.

⁴ é o capítulo que resolve o problema da não-linearidade física.

O capítulo *análise de seções no ELU* é uma adaptação do capítulo *determinação do estado de deformação da seção* para quando se analisa a seção isolada (no ELU de ruptura do concreto comprimido ou de deformação plástica excessiva da armadura tracionada). É tão importante quanto aquele.

No capítulo *análise da estabilidade*, são utilizados os métodos geral e do equilíbrio com os processos exato e do pilar padrão para se estudar a estabilidade de pilares⁵.

O *anexo A - métodos numéricos para o cálculo de zeros de funções* trata dos métodos numéricos para o cálculo de zeros de funções. Este anexo é fundamental para o entendimento dos algoritmos formulados nos capítulos 4, 5, 6 e 7.

No *apêndice 1 - listagem do sistema flexor*, são listadas todas as sub-rotinas computacionais que formam o SISTEMA FLEXOR, programa que materializa computacionalmente os capítulos 4, 5, 6 e 7, e que, portanto, efetua o cálculo de pilares esbeltos de concreto armado com seção qualquer solicitados por flexão composta oblíqua.

⁵ é o capítulo que resolve o problema da não-linearidade geométrica.

CAPÍTULO 2 - HIPÓTESES DE CÁLCULO

Neste trabalho, são assumidas as seguintes hipóteses:

2.1. MANUTENÇÃO DA SECÇÃO PLANA

Admite-se que as seções transversais ao eixo do pilar, inicialmente planas, permanecem planas e normais ao eixo deformado, e que esta condição é cumprida em qualquer estágio de carregamento. Esta hipótese implica em desprezar eventuais deformações por força cortante ou por momento torçor (flexo-torção).

2.2. MANUTENÇÃO DA FORMA DA SECÇÃO

Admite-se que a forma da seção não seja alterada com as deformações (por exemplo, por efeito de flambagem local ou de flexo-torção). Admite-se, também, que a seção se desloque transversalmente somente por translação, sem rotação.

2.3. SOLIDARIEDADE DOS MATERIAIS

Considera-se a existência de perfeita aderência entre o concreto e o aço, qualquer que seja o estágio de carregamento. Isto implica em que a armadura experimente a mesma deformação do concreto que lhe é adjacente.

2.4. INEXISTÊNCIA DE TENSÕES E DEFORMAÇÕES INICIAIS

Admite-se que não existam tensões e deformações iniciais ou residuais. Portanto, não são estudados elementos em concreto protendido.

2.5. PEQUENOS DESLOCAMENTOS

Ainda que sejam considerados neste trabalho efeitos de segunda ordem, admite-se como válida a teoria de pequenos deslocamentos. Em vista disso, pode-se adotar para a curvatura a expressão aproximada

$$\frac{1}{r} = \frac{d^2 w(z)}{dz^2} \quad (2.1)$$

onde $w(z)$ é o deslocamento transversal do eixo do pilar em uma posição genérica z .

2.6. CONSIDERAÇÃO DA FLUÊNCIA

Como a fluência ocorre sob a ação dos esforços permanentes de serviço, as tensões no concreto são suficientemente baixas para que se empregue a Teoria Linear da Fluência⁶, onde é admitida uma função ϕ de fluência independente da tensão aplicada. Nesse caso, sendo

ε_{c0} =deformação imediata do concreto;

ε_{cf} =deformação por fluência do concreto;

$\varepsilon_{c,total}$ =deformação total do concreto,

tem-se:

$$\varepsilon_{cf} = \phi \times \varepsilon_{c0} \quad (2.2)$$

$$\varepsilon_{c,total} = (1 + \phi) \times \varepsilon_{c0}$$

(2.3)

No caso de coexistirem cargas de curta e de longa duração, admite-se a realização do cálculo como se toda a carga fosse de longa duração, adotando-se para a função de fluência um valor equivalente efetivo, dado por

$$\phi_{ef} = \alpha_1 \times \alpha_2 \times \phi \quad (2.4)$$

onde

α_1 =fração da força normal que produz fluência;

α_2 =fração do momento fletor de primeira ordem que produz fluência.

Um valor usualmente adotado para ϕ é 2.

⁶ para informações complementares, consultar Araújo, J. M. (1985) - Modelo Linear para a consideração da Fluência do Concreto. Revista Estrutura, n.114, p.39-48, julho.

O efeito da fluência na metodologia de cálculo é discutido nos capítulos seguintes, à medida que se faz necessário.

2.7. TIPO DE SEÇÃO TRANSVERSAL

É admitida seção transversal de forma poligonal qualquer⁷. A disposição das barras da armadura também é arbitrária.

A seção é definida pelas coordenadas dos nós da poligonal, que devem ser numerados de 1 a N no sentido trigonométrico (anti-horário), com eventuais vazios no sentido contrário. O nó de chegada (N) deve coincidir com o nó de saída (1), fechando a poligonal. Desta maneira, fica automaticamente definida a numeração dos lados da poligonal: lado $i \rightarrow$ nó inicial= i , nó final= $i+1$ (uma poligonal de N nós tem N-1 lados). A figura 2.1 exemplifica:

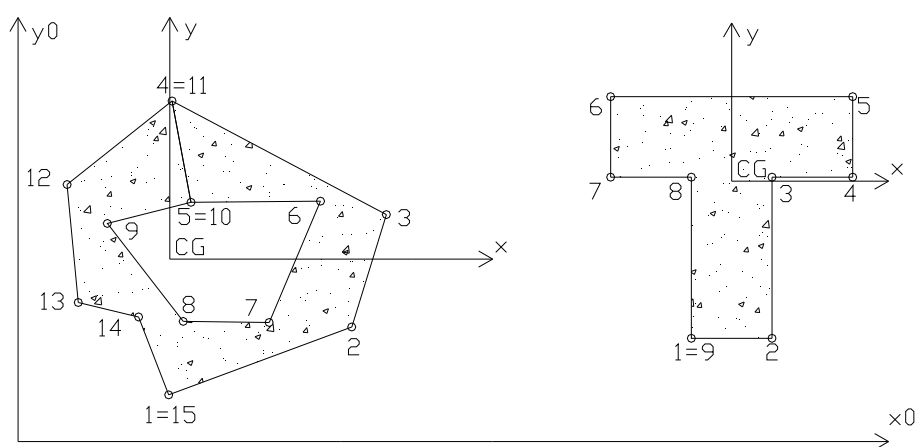


FIGURA 2.1 - Exemplos de poligonais.

A origem do sistema de coordenadas é convencionada como coincidente com o centro de gravidade (CG) da seção bruta de concreto. É em relação a este ponto que se consideram aplicados os esforços solicitantes e calculados os esforços resistentes (polo de redução dos esforços). Na verdade, pode-se colocar a origem do sistema de coordenadas em outro ponto qualquer que não o CG da seção bruta de concreto, mas deve-se ter em mente que os esforços solicitantes e os resistentes estarão vinculados a este ponto. Em se tratando

⁷ eventuais contornos curvos podem ser aproximados por uma poligonal lados pequenos.

de programa de computador, a seção pode ser inicialmente referida a um sistema qualquer e transladada para o CG automaticamente.

A armadura é definida pelas coordenadas e área de cada barra de aço. Por simplicidade, a área da armadura não é descontada da seção de concreto. Caso necessário, isto pode ser feito utilizando-se uma poligonal com vazios correspondentes às barras da armadura.

2.8. CONVENÇÃO DE SINAIS DE FORÇAS, TENSÕES, DEFORMAÇÕES, MOMENTOS FLETORES E DE CURVATURA

Neste trabalho, seguindo a maioria dos trabalhos correlatos, veja FORNI (1980), FRANÇA (1984), MENDES NETO (1991) e SANTOS (1994), é assumida a seguinte convenção de sinais de forças, tensões e de deformações:

- forças e tensões de compressão, deformações de encurtamento → sinal positivo;
- forças e tensões de tração, deformações de alongamento → sinal negativo.

A convenção de momentos fletores é assumida com base no conceito de excentricidade da força normal (ver fig. 2.2). Assim, considerando-se

M_x =momento fletor no plano que contém o eixo x;

M_y =momento fletor no plano que contém o eixo y;

e_x =excentricidade da força normal ao longo do eixo x;

e_y =excentricidade da força normal ao longo do eixo y;

N =força normal

tem-se:

$$M_x = N \cdot e_x \quad (2.5)$$

$$M_y = N \cdot e_y \quad (2.6)$$

Assim, os momentos são considerados positivos quando tendem a comprimir o 1º quadrante do sistema de coordenadas.

Alternativamente, os momentos fletores segundo os eixos x e y podem ser representados pelo módulo da resultante do momento fletor ($|M|$) e pela direção do seu plano de ação (β) (fig. 2.2):

$$|M| = \sqrt{M_x^2 + M_y^2} = N \cdot |e| \quad (2.7)$$

$$\beta = \arctan\left(\frac{M_y}{M_x}\right), \text{ com } (-\pi \leq \beta \leq +\pi) \quad (2.8)$$

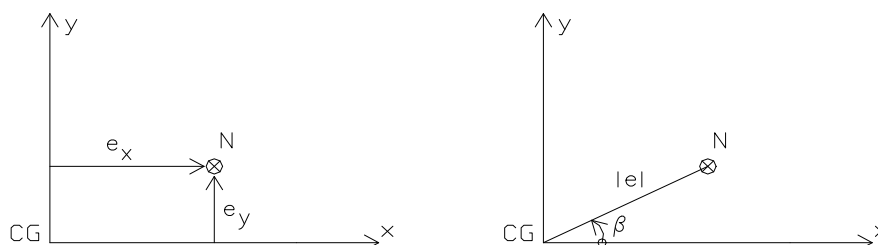


FIGURA 2.2 - Excentricidades da força normal.

Quanto à curvatura, é utilizado o conceito de curvatura majorada adimensional (θ). Seja um sistema de coordenadas Ouv rotacionado de α em relação ao sistema original Oxy ; admitindo-se o eixo u como paralelo à linha neutra da seção (fig. 2.3), define-se θ como:

$$\theta = 1000h_\alpha \frac{1}{r_\alpha} = 1000h_\alpha \frac{\varepsilon_2 / 1000 - \varepsilon_1 / 1000}{v_2 - v_1} = h_\alpha \frac{\varepsilon_2 - \varepsilon_1}{v_2 - v_1} \quad (2.9)$$

onde h_α é a altura da seção na direção de v , ε_1 e ε_2 são deformações (em ‰ e com sinal) em 2 pontos quaisquer da seção transversal e v_1 e v_2 são as ordenadas (sistema Ouv) destes pontos. Deste modo, θ é positivo se a seção deformada apresentar um maior encurtamento (ou menor alongamento) na porção acima do eixo u (sentido positivo do eixo v) em relação à porção abaixo do eixo u .

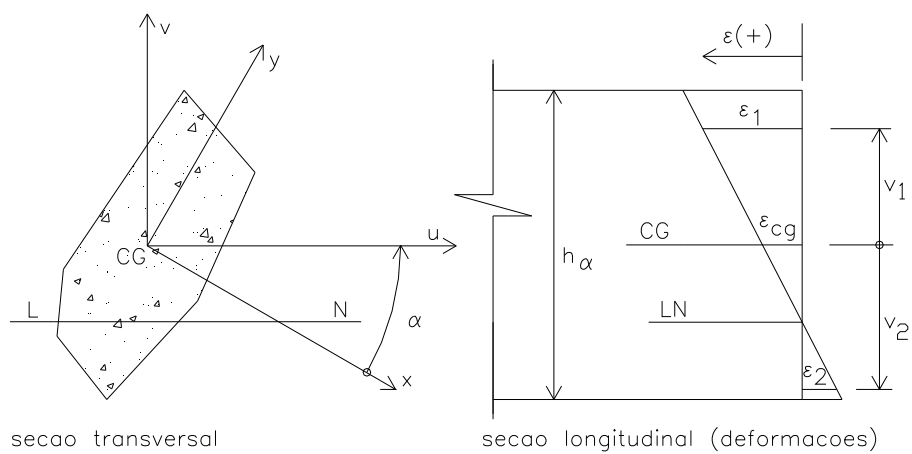


FIGURA 2.3 - Deformações na seção transversal - sistema de coordenadas Ouv.

CAPÍTULO 3 - EQUAÇÕES GERAIS

3.1. EQUAÇÕES DE EQUILÍBRIO

Quando uma seção de concreto armado é solicitada por FCO (força normal solicitante N_s , e momento fletor solicitante, dado por suas componentes M_{sx} e M_{sy}), nela surgem deformações e tensões normais. Estas tensões formam os esforços resistentes, que equilibram os esforços solicitantes de modo a garantir o equilíbrio da seção.

Da condição de igualdade entre a força normal resistente (N_r) e a força normal solicitante (N_s), vem:

$$N_r = N_s \quad (3.1)$$

Enquanto N_s é um dado do problema, N_r é um valor a ser calculado. Este, é composto de duas parcelas: a força normal resistente fornecida pelo aço (N_{rs}) e a força normal resistente fornecida pelo concreto (N_{rc}). Assim, tem-se:

$$N_r = N_{rs} + N_{rc} \quad (3.2)$$

com:

$$N_{rs} = \sum_{i=1}^{n^{\circ}\text{barras}} N_{rs,i} = \sum_{i=1}^{n^{\circ}\text{barras}} A_{s,i} \cdot \sigma_{s,i} \quad (3.3)$$

$$N_{rc} = \int_{A_{cc}} \sigma_c \cdot dA \quad (3.4)$$

Da condição de igualdade entre os momentos fletores resistentes (M_{rx} e M_{ry}) e os momentos fletores solicitantes (M_{sx} e M_{sy}), vem:

$$M_{rx} = M_{sx} \quad (3.5)$$

$$M_{ry} = M_{sy} \quad (3.6)$$

Da mesma forma que a força normal resistente, os momentos fletores resistentes são valores a serem calculados, e são compostos de duas parcelas: os momentos fletores resistentes fornecido pelo aço (M_{rxs} e M_{rys}) e os momentos fletores resistentes fornecido pelo concreto comprimido (M_{rxc} e M_{ryc}). Assim, tem-se:

$$M_{rx} = M_{rxs} + M_{rxc} \quad (3.7)$$

com:

$$M_{rxs} = \sum_{i=1}^{n^{\circ}\text{barras}} M_{rxs,i} = \sum_{i=1}^{n^{\circ}\text{barras}} A_{s,i} \cdot \sigma_{s,i} \cdot x_{s,i} \quad (3.8)$$

$$M_{rxc} = \int_{A_{cc}} \sigma_c \cdot x \cdot dA \quad (3.9)$$

e,

$$M_{ry} = M_{rys} + M_{ryc} \quad (3.10)$$

com:

$$M_{rys} = \sum_{i=1}^{n^{\circ}\text{barras}} M_{rys,i} = \sum_{i=1}^{n^{\circ}\text{barras}} A_{s,i} \cdot \sigma_{s,i} \cdot y_{s,i} \quad (3.11)$$

$$M_{ryc} = \int_{A_{cc}} \sigma_c \cdot y \cdot dA \quad (3.12)$$

Nas equações acima,

$A_{s,i}$ = área da seção transversal da barra de aço genérica i ;

$x_{s,i}, y_{s,i}$ = coordenadas da barra de aço i ;

$\sigma_{s,i}$ = tensão na barra de aço i ;

σ_c = tensão no concreto no ponto de coordenadas x, y da seção transversal;

A_{cc} = área da porção comprimida da seção transversal de concreto.

O cálculo dos valores definidos pelas equações 3.3, 3.4, 3.8, 3.9 3.11 e 3.12 é o objeto de estudo do capítulo 4.

3.2. EQUAÇÕES DE COMPATIBILIDADE DE DEFORMAÇÕES

3.2.1. EQUAÇÃO DE COMPATIBILIDADE GERAL

A hipótese de seções planas implica em uma distribuição linear de deformações ao longo da seção transversal (equação de plano). Assim, é necessário o conhecimento de 3 parâmetros para a determinação completa do estado de deformação da seção.

Alguns parâmetros que podem ser utilizados para compor o conjunto de 3 parâmetros necessários são:

- α : ângulo entre o eixo x e a direção da linha neutra (LN), medido no sentido trigonométrico (anti-horário);
- deformação em um ponto da seção transversal;

- β_v : profundidade adimensional da linha neutra $\rightarrow \beta_v = \frac{V_{LN}}{h_\alpha}$;
- θ : curvatura majorada adimensional $\rightarrow \theta = 1000 \frac{h_\alpha}{r}$

Alguns conjuntos de 3 parâmetros disponíveis para utilização são:

- a) 3 deformações em 3 pontos diferentes da seção transversal
- b) α e 2 deformações em 2 pontos diferentes da seção transversal;
- c) α , 1 deformação em um ponto qualquer da seção transversal e β_v ;
- d) α , 1 deformação em um ponto qualquer da seção transversal e θ ;
- e) α , β_v e θ .

Os conjuntos geralmente utilizados nos trabalhos correlatos são o ‘d’ e o ‘e’. Neste trabalho, optou-se pela utilização do conjunto ‘d’, com a deformação sendo referida à origem do sistema de coordenadas (onde se supõe estar o CG da seção), ε_{CG} . Esta escolha baseou-se no fato de ser a *sensibilidade* do parâmetro β_v muito variável. Explica-se: quando β_v varia, por exemplo, de 0 a 0,1, os esforços resistentes variam muito; mas quando se varia β_v , por exemplo, de 2 a 2,1, os esforços resistentes quase não variam; além disso, a faixa de variação de β_v é, teoricamente, de $-\infty$ a $+\infty$. Estes aspectos dificultam a implementação computacional da teoria em estudo, daí a opção pelo parâmetro ε_{CG} , que também apresenta os problemas descritos, mas em menor intensidade.

Como se viu, o estado de deformação da seção é representado pela equação de plano (3 parâmetros). Porém, computacionalmente, é sempre possível rotacionar a seção para que o sistema de coordenadas tenha o eixo das abscissas paralelo à linha neutra (sistema Ouv). Neste sistema, a equação de plano degenera em equação de reta (2 parâmetros). Seja, então, uma seção referida ao sistema Ouv, conforme a fig. 3.1:

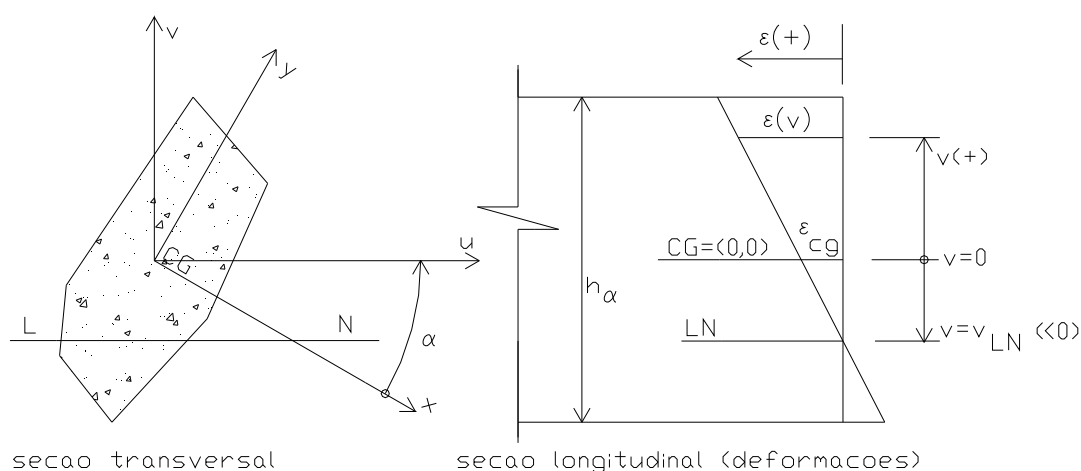


FIGURA 3.1 - Estado de deformação de uma seção transversal.

o seu estado de deformação é dado pela equação de uma reta, ou seja:

$$\varepsilon = A + B \cdot v \quad (3.13)$$

$$\text{em } v=0, \varepsilon = \varepsilon_{CG} \rightarrow \varepsilon_{CG} = A + B \times 0 \therefore A = \varepsilon_{CG} \quad (3.14)$$

$$\text{em } v = v_{LN}, \varepsilon = 0 \rightarrow 0 = A + B \times v_{LN} \therefore B = -\varepsilon_{CG} / v_{LN} \quad (3.15)$$

mas, da definição de θ (ver item 2.8), vem:

$$\theta = h_{\alpha} \frac{\varepsilon_2 - \varepsilon_1}{v_2 - v_1} = h_{\alpha} \frac{\varepsilon_{CG} - 0}{0 - v_{LN}} = h_{\alpha} \frac{-\varepsilon_{CG}}{v_{LN}} \quad (3.16)$$

da comparação de 3.15 com 3.16, vem:

$$B = \theta / h_{\alpha} \quad (3.17)$$

finalmente, de 3.14 e 3.17 em 3.13, vem a equação geral de compatibilidade de deformações:

$$\varepsilon = \varepsilon_{CG} + \frac{\theta}{h_{\alpha}} \cdot v \quad (3.18)$$

onde: ε = deformação (em ‰ e com sinal) da fibra de ordenada v da seção transversal;

v = ordenada (sist. Ouv) da fibra onde se deseja calcular ε ;

ε_{CG} = deformação (em ‰ e com sinal) da origem do sistema de coordenadas (onde se supõe estar o CG da seção; isto, porém, não é obrigatório);

θ = curvatura majorada adimensional ($1000h_{\alpha}/r$);

h_{α} = altura da seção, medida na direção perpendicular à da linha neutra (eixo v).

3.2.2. EQUAÇÃO DE COMPATIBILIDADE NO ELU

O Estado Limite Último de ruptura do concreto comprimido ou de deformação plástica excessiva da armadura tracionada (ou simplesmente ELU) é caracterizado pelos domínios de deformação da NB1/78. Conforme observado por SANTOS (1983), são 3 as condições que levam uma seção ao ELU (são os *pólos de ruína*). Já considerando o efeito da fluência, estas condições são (fig. 3.2):

- seção inteiramente comprimida:
 - pólo 1: encurtamento de $2(1+\phi)$ (em ‰) na fibra situada em $3h_\alpha/7$ a partir da fibra mais encurtada;
- seção não inteiramente comprimida:
 - pólo 2: encurtamento máximo do concreto igual a $3,5(1+\phi)$ (em ‰);
 - pólo 3: alongamento máximo da armadura igual a -10% .

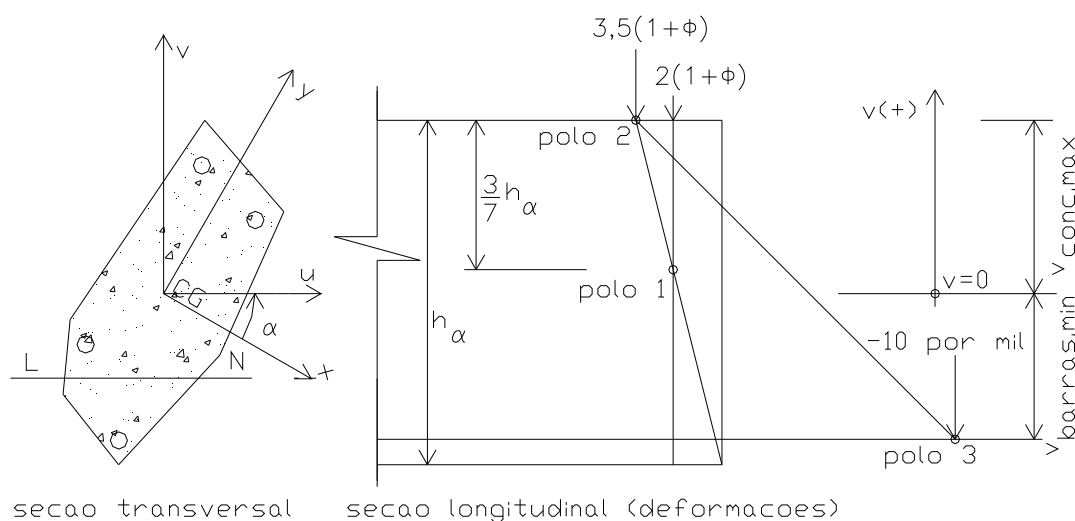


FIGURA 3.2 - Pólos de ruína do ELU.

Como se viu no item 3.2.1, é necessário o conhecimento de 3 parâmetros (α , ϵ_{CG} e θ) para a completa determinação do estado de deformação da seção. Porém, a consideração de a seção estar no ELU por si só já se constitui num parâmetro dado, o que, portanto, diminui para 2 o número de parâmetros necessários para se escrever a equação de compatibilidade. Estando a seção referida ao sistema Ouv (sistema rotacionado de α em relação ao sistema Oxy), é necessário somente 1 parâmetro. Analisando-se, então, a equação 3.18, deduz-se que, no ELU, ϵ_{CG} e θ não são independentes um do outro, mas que um é função do outro.

Seja uma seção referida ao sistema Ouv. Em se tratando de ELU, há duas perguntas a serem respondidas:

- 1) Dado ε_{CG} , pergunta-se qual o θ que coloca a seção no ELU: trata-se de encontrar $\theta = f(\varepsilon_{CG})$. Respondido isto, basta substituir os dois parâmetros (θ e ε_{CG}) na equação 3.18 para que se tenha a equação de compatibilidade no ELU. O problema assim resolvido é necessário para o dimensionamento de seções no ELU.
- 2) Dado θ e ε_{CG} , pergunta-se se a seção ultrapassou ou não o ELU (atingiu ou não a ruína): trata-se de encontrar $\varepsilon_{CG,mínimo} = f(\theta)$ e $\varepsilon_{CG,máximo} = f(\theta)$. Para considerar que a seção atingiu a ruína, é suficiente que $\varepsilon_{CG} > \varepsilon_{CG,máximo}$ ou que $\varepsilon_{CG} < \varepsilon_{CG,mínimo}$. O problema assim resolvido é necessário para o estudo da estabilidade de pilares (informa se determinada seção rompeu ou não).

Resolvamos a 1ª questão, $\theta = f(\varepsilon_{CG})^8$.

Admitindo-se que a seção esteja no ELU e, ainda, que o estado de deformação passe pelo pólo 1 (ver figura 3.2), tem-se, aplicando-se a eq. 3.18:

$$\varepsilon = \varepsilon_{CG} + \frac{\theta}{h_\alpha} \cdot v \rightarrow 2(1 + \phi) = \varepsilon_{CG} + \frac{\theta}{h_\alpha} (v_{CONCR,MAX} - \frac{3}{7} h_\alpha)$$

$$\therefore \theta_1 = \frac{h_\alpha (2(1 + \phi) - \varepsilon_{CG})}{v_{CONCR,MAX} - \frac{3}{7} h_\alpha} \quad (3.19)$$

Admitindo-se que a seção esteja no ELU e, ainda, que o estado de deformação passe pelo pólo 2 (ver figura 3.2), tem-se, aplicando-se a eq. 3.18:

$$\varepsilon = \varepsilon_{CG} + \frac{\theta}{h_\alpha} \cdot v \rightarrow 3,5(1 + \phi) = \varepsilon_{CG} + \frac{\theta}{h_\alpha} (v_{CONCR,MAX})$$

$$\therefore \theta_2 = \frac{h_\alpha (3,5(1 + \phi) - \varepsilon_{CG})}{v_{CONCR,MAX}} \quad (3.20)$$

Admitindo-se que a seção esteja no ELU e, ainda, que o estado de deformação passe pelo pólo 3 (ver figura 3.2), tem-se, aplicando-se a eq. 3.18:

$$\varepsilon = \varepsilon_{CG} + \frac{\theta}{h_\alpha} \cdot v \rightarrow -10 = \varepsilon_{CG} + \frac{\theta}{h_\alpha} (v_{BARRAS,MIN})$$

$$\therefore \theta_3 = \frac{h_\alpha (-10 - \varepsilon_{CG})}{V_{BARRAS,MIN}} \quad (3.21)$$

Analisando-se a figura 3.2 mais a fundo e supondo-se $\theta \geq 0$, $v_{barras,min} < 0$ e $v_{conc,max} - 3h_\alpha/7 > 0$, deduz-se que a resposta à pergunta 1 ($\theta = f(\varepsilon_{CG})$) é:

$$\theta \leq \begin{cases} \theta_1 \text{ (eq.3.19)} \\ \theta_2 \text{ (eq.3.20)} \\ \theta_3 \text{ (eq.3.21)} \end{cases} \quad (3.22)$$

obs.: caso o θ_1 calculado resulte negativo, ele não deve entrar na eq. 3.22; o mesmo ocorre com θ_2 e θ_3 ; caso $v_{barras,min} > 0$ ou $v_{conc,max} - 3h_\alpha/7 < 0$, basta realizar uma translação temporária da seção até que estas condições sejam atendidas.

Passemos agora à 2ª questão, $\varepsilon_{CG,mínimo} = f(\theta)$ e $\varepsilon_{CG,máximo} = f(\theta)$ ⁹.

Admitindo-se que a seção esteja no ELU e, ainda, que o estado de deformação passe pelo pólo 1 (ver figura 3.2), tem-se, aplicando-se a eq. 3.18:

$$\begin{aligned} \varepsilon &= \varepsilon_{CG} + \frac{\theta}{h_\alpha} \cdot v \rightarrow 2(1 + \phi) = \varepsilon_{CG} + \frac{\theta}{h_\alpha} (v_{CONCR,MAX} - \frac{3}{7} h_\alpha) \\ \therefore \varepsilon_{CG,1} &= -\frac{\theta}{h_\alpha} (v_{CONCR,MAX} - \frac{3}{7} h_\alpha) + 2(1 + \phi) \end{aligned} \quad (3.23)$$

Admitindo-se que a seção esteja no ELU e, ainda, que o estado de deformação passe pelo pólo 2 (ver figura 3.2), tem-se, aplicando-se a eq. 3.18:

$$\begin{aligned} \varepsilon &= \varepsilon_{CG} + \frac{\theta}{h_\alpha} \cdot v \rightarrow 3,5(1 + \phi) = \varepsilon_{CG} + \frac{\theta}{h_\alpha} (v_{CONCR,MAX}) \\ \therefore \varepsilon_{CG,2} &= +\frac{\theta}{h_\alpha} (v_{CONCR,MAX}) + 3,5(1 + \phi) \end{aligned} \quad (3.24)$$

Admitindo-se que a seção esteja no ELU e, ainda, que o estado de deformação passe pelo pólo 3 (ver figura 3.2), tem-se, aplicando-se a eq. 3.18:

⁸ a materialização computacional desta questão é a sub-rotina “e_CGxTETA_ELU”. constante do Apêndice 1.

⁹ a materialização computacional desta questão é a sub-rotina “TETA_eCG_ELU”, constante do Apêndice 1.

$$\varepsilon = \varepsilon_{CG} + \frac{\theta}{h_\alpha} \cdot v \rightarrow -10 = \varepsilon_{CG} + \frac{\theta}{h_\alpha} (v_{BARRAS,MIN})$$

$$\therefore \varepsilon_{CG,3} = -\frac{\theta}{h_\alpha} (v_{BARRAS,MIN}) - 10 \quad (3.25)$$

Analisando-se a figura 3.2 mais a fundo, deduz-se que a resposta à pergunta 2 ($\varepsilon_{CG,mínimo} = f(\theta)$ e $\varepsilon_{CG,máximo} = f(\theta)$) é:

$$\varepsilon_{CG,maximo} \leq \begin{cases} \varepsilon_{CG,1} \text{ (eq.3.23)} \\ \varepsilon_{CG,2} \text{ (eq.3.24)} \end{cases} \quad (3.26)$$

$$\varepsilon_{CG,minimo} = \varepsilon_{CG,3} \text{ (eq.3.25)} \quad (3.27)$$

3.3. EQUACÕES CONSTITUTIVAS

3.3.1. AÇO

Neste trabalho, são admitidos os aços caracterizados pela NB1/78 - aços classe A ou classe B. A materialização computacional deste item é a sub-rotina “TENSAO_ACO”, constante do Apêndice 1.

Os diagramas tensão-deformação de cálculo são aqueles definidos na NB1/78 para os aços classe A e classe B (veja fig. 3.3). Assim,

a) aço classe A:

$$\varepsilon_{yd} = 1000 \frac{fy}{Es} \quad (3.28)$$

$$0 \leq |\varepsilon_s| \leq \varepsilon_{yd} \rightarrow \sigma_s = \frac{Es \cdot \varepsilon_s}{1000} \quad (3.29)$$

$$|\varepsilon_s| \geq \varepsilon_{yd} \rightarrow \sigma_s = \frac{\varepsilon_s}{|\varepsilon_s|} \cdot fy \quad (3.30)$$

b) aço classe B:

$$\varepsilon_{yd} = 1000 \frac{fy}{Es} + 2 \quad (3.31)$$

$$|\varepsilon_s| \leq 0,7 \times 1000 fy / Es \rightarrow \sigma_s = \frac{Es \cdot \varepsilon_s}{1000} \quad (3.32)$$

$$700fy/Es < |\epsilon_s| < \epsilon_{yd} \rightarrow \sigma_s = \frac{\epsilon_s}{|\epsilon_s|} fy \left\{ 0,7 - \frac{45fy}{2} \left[\frac{1}{Es} - \sqrt{\frac{1}{Es^2} + \frac{4}{45fy} \left(\frac{|\epsilon_s|}{1000fy} - \frac{0,7}{Es} \right)} \right] \right\}$$

(3.33)

$$|\epsilon_s| \geq \epsilon_{yd} \rightarrow \sigma_s = \frac{\epsilon_s}{|\epsilon_s|} \cdot fy$$

(3.34)

onde:

ϵ_s = deformação (em % e com sinal) da barra de aço;

σ_s = tensão na barra de aço (com sinal);

E_s = módulo de deformação longitudinal da armadura (usualmente, $E_s=210000\text{Mpa}$);

f_y = tensão de escoamento da armadura; usualmente, faz-se $f_y=f_{yd}=f_{yk}/\gamma_s$, com $\gamma_s=1,15$;

porém, no estudo da estabilidade, pode-se assumir $\gamma_s=1,0$ (veja observação a seguir).

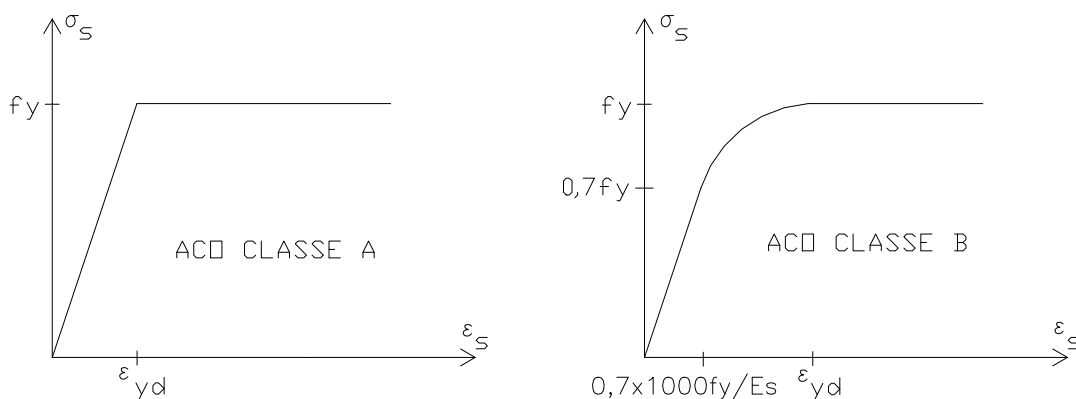


FIGURA 3.3 - Diagramas tensão-deformação para os aços.

3.3.2. CONCRETO

Admite-se que o diagrama tensão-deformação do concreto seja formado por uma seqüência de polinômios de graus qualquer, conforme exemplifica a figura 3.4. Assim, pode ser obtido praticamente qualquer tipo de diagrama tensão-deformação, inclusive diagramas que admitem resistência do concreto à tração.

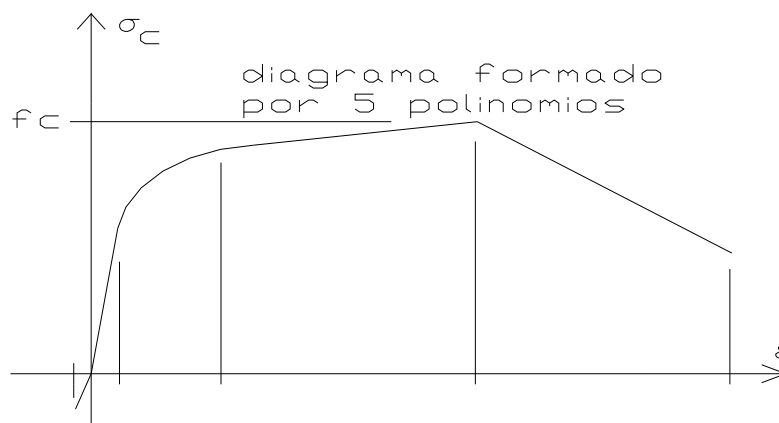


FIGURA 3.4 - Exemplo de diagrama tensão-deformação para o concreto.

Cada curva polinomial é definida por:

$$\sigma_c(\varepsilon) = f_c \sum_{k=0}^m C_k \cdot \varepsilon^k = f_c(C_0 \cdot \varepsilon^0 + C_1 \cdot \varepsilon^1 + C_2 \cdot \varepsilon^2 + \dots + C_m \cdot \varepsilon^m) \quad (3.35)$$

onde:

f_c = tensão máxima de compressão de todo o diagrama tensão-deformação do concreto, não só do polinômio em questão (veja fig. 3.4); obs.: ver a seguir a consideração sobre o valor de f_c a se utilizar;

m = grau do polinômio (reta $\rightarrow m=0$; parábola do 2º grau $\rightarrow m=2$, etc.);

C_k = k 'ésimo coeficiente do polinômio; $k \rightarrow 0$ a m ; os coeficientes do polinômio devem representar a curva como se todo o carregamento fosse de curta duração (sem o efeito da fluência);

ε = deformação (em ‰ e com sinal) da fibra de concreto onde se deseja calcular a tensão.

O diagrama tensão-deformação completo do concreto é definido por:

- N_m = número de polinômios que compõem o diagrama tensão-deformação adotado para o concreto;
- VETOR_m = vetor ($N_m \times 1$) que contém os graus m dos N_m polinômios;
- MATRIZ_C = matriz ($N_m \times$ maior m) que contém os coeficientes C_k dos N_m polinômios; a linha i desta matriz contém os coeficientes do polinômio i ; estes coeficientes não devem incluir o efeito da fluência;
- MATRIZ_e_LIMITES = matriz ($N_m \times 2$) que contém as deformações (em ‰ e com sinal) onde inicia e termina cada uma das N_m curvas polinomiais; a linha i desta

matriz contém as deformações onde inicia e termina a atuação do polinômio i ; estas deformações não devem incluir o efeito da fluência;

Para o diagrama tensão-deformação Parábola-Retângulo definido pela NB1/78, fig. 3.5, tem-se:

- $N_m = 2$
- $VETOR_m = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$
- $MATRIZ_C = \begin{bmatrix} 0 & 1 & -0,25 \\ 1 & 0 & 0 \end{bmatrix}$
- $MATRIZ_e_LIMITES = \begin{bmatrix} 0 & 2 \\ 2 & 3,5 \end{bmatrix}$

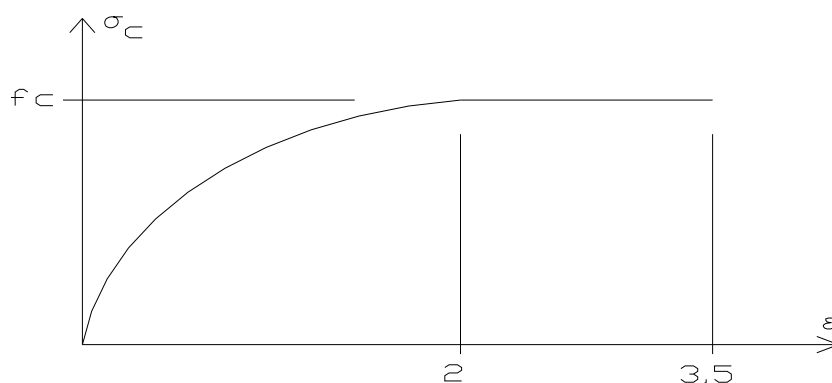


FIGURA 3.5 - Diagrama tensão-deformação Parábola-Retângulo.

Quanto aos efeitos de carregamentos de longa duração (fluência), observa-se que a equação 3.35, a $MATRIZ_C$ e a $MATRIZ_e_LIMITES$ não consideram o efeito da fluência. O efeito da fluência equivale à uma transformação afim (de razão ϕ) paralela ao eixo das abscissas no diagrama σ - ϵ do concreto¹⁰ (fig. 3.6).

¹⁰ admitindo-se a Teoria Linear de Fluência, conforme exposto na seção 2.6.

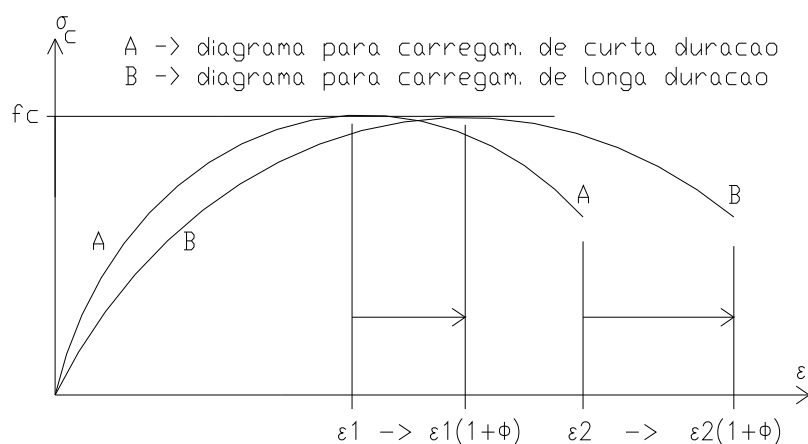


FIGURA 3.6 - Diagramas σ - ε para cargas de curta e de longa duração.

Por conta da transformação afim no diagrama σ - ε do concreto - devida ao efeito da fluência - a equação 3.35 deve ser modificada para:

$$\sigma_c(\varepsilon) = f_c \sum_{k=0}^m \frac{C_k}{(1+\phi)^k} \varepsilon^k = f_c \left(\frac{C_0}{(1+\phi)^0} \varepsilon^0 + \frac{C_1}{(1+\phi)^1} \varepsilon^1 + \frac{C_2}{(1+\phi)^2} \varepsilon^2 + \dots + \frac{C_m}{(1+\phi)^m} \varepsilon^m \right) \quad (3.36)$$

Analogamente, as deformações onde inicia e termina a atuação de cada curva polinomial no diagrama σ - ε do concreto são, sob o efeito da fluência, os valores da MATRIZ_e_LIMITES multiplicados por $(1+\phi)$.

Quanto ao valor a se empregar para f_c , no caso do estudo da estabilidade ainda não há consenso.

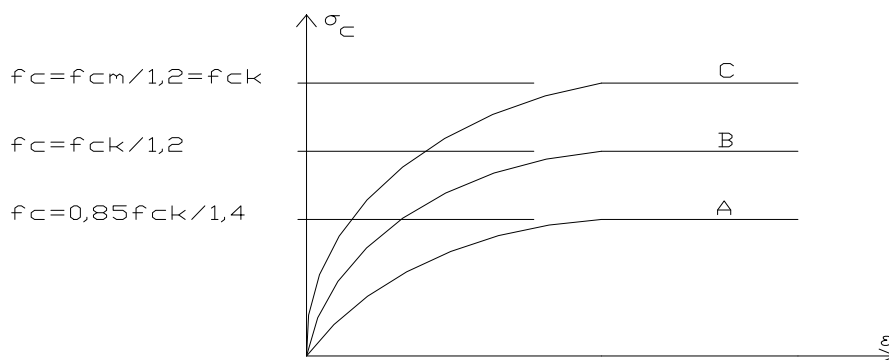


FIGURA 3.7 - Possíveis valores de f_c para o estudo da estabilidade.

O f_c do diagrama A (fig. 3.7) é o recomendado pela NB1/78. Ele é adequado para o dimensionamento de seções no ELU. Porém, no estudo da estabilidade, este valor leva a uma

estimativa exagerada da deformabilidade da estrutura. FUSCO (1981), sugere a adoção do f_c dado no diagrama B para pilares esbeltos ($\lambda > 80$) e para os não contraventados, permitindo o f_c do diagrama C para os pilares medianamente esbeltos ($\lambda < 80$). Analogamente, no estudo da estabilidade, pode ser usado, para o aço, o valor f_{yk} ao invés de f_{yd} .

CAPÍTULO 4 - CÁLCULO DOS ESFORÇOS RESISTENTES DA SEÇÃO

Este capítulo é de fundamental importância, uma vez que o cálculo dos esforços resistentes é um passo básico na resolução de qualquer problema de dimensionamento de peças de concreto armado¹¹.

4.1. APRESENTAÇÃO DO PROBLEMA

O objeto de estudo deste capítulo é o cálculo dos esforços resistentes (N_r , M_{rx} e M_{ry}) de uma seção transversal. Deve-se ter em mente, então, que são dados do problema: seção transversal definida geométrica¹² e mecanicamente¹³ e o estado de deformação (α , θ e ε_{CG}) a que a seção está sujeita. Esquemáticamente, tem-se:

$$\left\{ \begin{array}{c} \text{s e c a o} \\ \alpha \\ \theta \\ \varepsilon_{CG} \end{array} \right\} \Rightarrow \left\{ \begin{array}{c} N_r \\ M_{rx} \\ M_{ry} \end{array} \right\}$$

Para facilitar o processo de cálculo dos esforços resistentes, ao longo deste capítulo a seção é sempre referida a um sistema de coordenadas Ouv, no qual o eixo das abscissas (u) é paralelo à LN (fig. 4.1). Este sistema é obtido através de uma rotação de α no sentido trigonométrico no sistema de coordenadas original (Oxy). A conversão do sistema Oxy para o sistema Ouv é feita com:

$$\begin{aligned} u &= x \cdot \cos(\alpha) + y \cdot \text{sen}(\alpha) \\ v &= -x \cdot \text{sen}(\alpha) + y \cdot \cos(\alpha) \end{aligned} \tag{4.1}$$

¹¹ A materialização computacional deste item é a sub-rotina “ESFORCOS_SECAO”, constante do Apêndice 1.

¹² Poligonal de concreto, posição e área das barras da armadura.

¹³ Equações constitutivas do concreto e do aço.

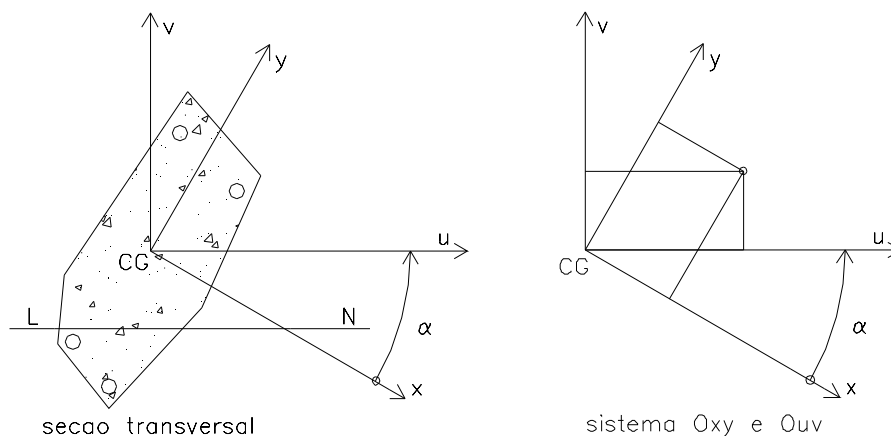


FIGURA 4.1 - Sistema de coordenadas Ouv.

Os esforços resistentes calculados com a seção referida ao sistema Ouv são também relacionados a este sistema (N_r , M_{ru} e M_{rv}). Para se obter os esforços resistentes no sistema Oxy, faz-se:

$$\begin{aligned} M_{rx} &= M_{ru} \cdot \cos(\alpha) - M_{rv} \cdot \sin(\alpha) \\ M_{ry} &= M_{ru} \cdot \sin(\alpha) + M_{rv} \cdot \cos(\alpha) \end{aligned} \quad (4.2)$$

Conforme observado no item 3.1, os esforços resistentes de uma seção podem ser considerados como a soma de duas parcelas: contribuição resistente da armadura (N_{rs} , M_{rus} e M_{rvs}) e contribuição resistente do concreto (N_{rc} , M_{ruc} e M_{rvc}). O cálculo destas contribuições é o objeto de estudo dos itens seguintes.

4.2. CÁLCULO DA CONTRIBUIÇÃO RESISTENTE DO AÇO

A armadura de uma seção transversal é definida pelas coordenadas (sist. Ouv) e área de cada barra de aço.

O cálculo da contribuição resistente do aço é feito através das equações 3.3, 3.8 e 3.11 (com os índices x e y trocados por u e v). Nestas equações, $\sigma_{s,i}$ e ε_s são dados de entrada; $\sigma_{s,i}$ é calculado utilizando-se as equações 3.28 a 3.34 e ε_s é calculado utilizando-se a equação 3.18.

Marcha de cálculo:

- 1) $i=1$ (i é o número da barra de aço)

$$2) \varepsilon_{s,i} = \varepsilon_{CG} + \frac{\theta}{h_{\alpha}} \cdot v_{s,i} \quad (\text{eq. 3.18})$$

3) $\sigma_{s,i} \rightarrow$ equações 3.28 a 3.34

$$4) N_{rs,i} = A_{s,i} \cdot \sigma_{s,i}$$

$$5) M_{rus,i} = A_{s,i} \cdot \sigma_{s,i} \cdot u_{s,i}$$

$$6) M_{rvs,i} = A_{s,i} \cdot \sigma_{s,i} \cdot v_{s,i}$$

$$7) N_{rs} = N_{rs} + N_{rs,i}$$

$$8) M_{rus} = M_{rus} + M_{rus,i}$$

$$9) M_{rvs} = M_{rvs} + M_{rvs,i}$$

10) $i=i+1$

11) se $i \leq N_{\text{BARRAS DE AÇO}} \rightarrow$ volta-se ao passo 2

4.3. CÁLCULO DA CONTRIBUIÇÃO RESISTENTE DO CONCRETO

4.3.1. COMENTÁRIOS INICIAIS

O cálculo dos esforços resistentes do concreto é um dos principais problemas a ser vencido em um estudo de dimensionamento de peças de concreto armado. Trata-se do cálculo das integrais de superfície dadas nas equações 3.4, 3.9 e 3.12, que, adaptadas ao sistema Ouv, se tornam:

$$N_{rc} = \int_{A_{cc}} \sigma_c \cdot dA \quad (4.3)$$

$$M_{ruc} = \int_{A_{cc}} \sigma_c \cdot u \cdot dA \quad (4.4)$$

$$M_{rvc} = \int_{A_{cc}} \sigma_c \cdot v \cdot dA \quad (4.5)$$

É admitida seção transversal poligonal de forma qualquer, definida como indica o item 2.7, e diagrama σ - ε (tensão-deformação) do concreto formado por uma seqüência de polinômios de graus arbitrários, definidos como indica o item 3.2.2.

4.3.2. PRINCIPAIS MÉTODOS DE INTEGRAÇÃO DAS TENSÕES AO LONGO DA SEÇÃO TRANSVERSAL

Os processos mais conhecidos para a resolução das integrais relativas ao concreto são:

- 1) PROCESSO DAS MALHAS
- 2) PROCESSO DA INTEGRAÇÃO NUMÉRICA
- 3) PROCESSO DA POLIGONAL: A-INTEGRAÇÃO NUMÉRICA
B-INTEGRAÇÃO ANALÍTICA

No primeiro processo, divide-se a seção transversal num grande número de elementos (fig. 4.2), de modo a se poder considerar, em cada elemento, a tensão como constante. Assim, as integrais das equações 4.3, 4.4 e 4.5 podem ser aproximadas por somatórias. É um processo largamente difundido; veja, por exemplo, CABRÉ (1972), MONTOYA et al. (1973) e MARINO (1978). Apesar disso, apresenta alguns inconvenientes, conforme salienta SANTOS (1981): nas vizinhanças da LN e do contorno da seção os elementos podem ficar mutilados, gerando imprecisão; há necessidade de dividir a seção em um grande número de elementos para se conseguir resultados com boa precisão, o que representa uma grande quantidade de cálculos; há necessidade de cuidados especiais para que as deformações máximas em vértices sejam acusadas; há grande possibilidade de erro na entrada de dados. Por isso, o Processo das Malhas não é o utilizado neste trabalho.

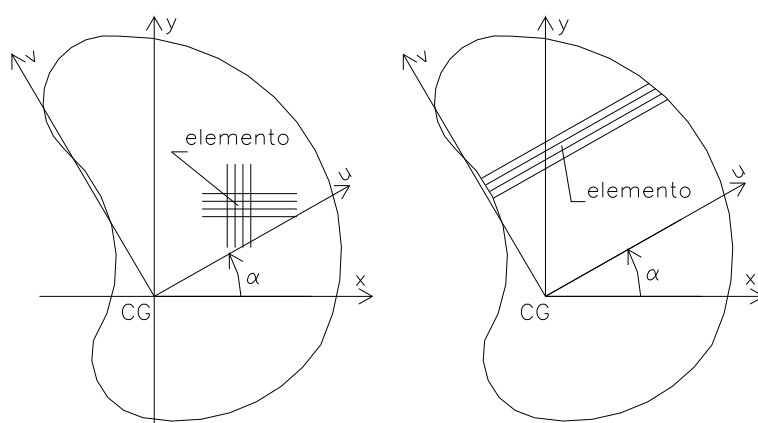


FIGURA 4.2 - Processo das Malhas - tipos de elementos.

O segundo processo - Processo da Integração Numérica - consiste em se resolver as integrais 4.3, 4.4 e 4.5 numericamente (veja, por exemplo, VENTURINI & BORTOLIN (1992)). Define-se uma área retangular que circunscreve a seção e nela distribui-se pontos de

Gauss; considera-se que a integral da seção real pode ser representada pela integral sobre o retângulo (considera-se tensão nula nos pontos externos à seção real); utilizando-se um esquema de integração numérica do tipo gausseano, por exemplo, calculam-se as integrais numéricas. Como se trata de integração numérica, adapta-se facilmente a qualquer digrama σ - ε adotado para o concreto. A precisão do método depende do número de pontos de Gauss utilizados; assim, este método pode ser muito preciso, mas isto acarreta grande quantidade de cálculos. Não é o processo utilizado neste trabalho.

O terceiro processo - Processo da Poligonal - é muito difundido; veja, por exemplo, WERNER (1974), FORNI (1980), FRANÇA (1984), MENDES NETO (1991), SANTOS (1994) e CAMPOS (1994). Neste processo, as integrais de superfície, que se estendem sobre a área de concreto, são transformadas em integrais curvilíneas, estendidas ao longo do contorno da seção. Isto é conseguido através da aplicação do Teorema de GAUSS-GREEN. A integral de linha assim obtida, pode ser resolvida de duas maneiras:

- a) utilizando-se um método numérico de integração, como, por exemplo, o Método de Quadratura de Gauss; uma vantagem da integração numérica é a sua adaptação a qualquer diagrama σ - ε adotado; como desvantagens, ele importa em uma grande quantidade de cálculos e não fornece a resposta exata do problema (embora possa ser muito preciso);
- b) desenvolvendo-se uma integração de forma fechada para determinado tipo de diagrama σ - ε do concreto (integração analítica); a principal vantagem deste tipo de solução é que a resposta obtida é, teoricamente, exata (veja o item 4.3.4); a principal desvantagem é o fato de a dedução da formulação exigir que se pré-defina o tipo de diagrama σ - ε para o concreto, o que não ocorre quando se utiliza a integração numérica.

Neste trabalho, optou-se pela utilização do Processo da Poligonal com Integração Analítica. Tal escolha baseou-se no fato de tal método fornecer respostas teoricamente exatas e, além disso, não implicar em excessiva quantidade de cálculos (menor tempo de processamento computacional). A principal desvantagem de tal método - diagrama σ - ε pré-definido - é amenizada pela adoção de diagrama σ - ε formado por uma seqüência de curvas polinomiais de graus arbitrários, que pode simular praticamente qualquer tipo de diagrama σ - ε que se queira.

4.3.3. PROCESSO DA POLIGONAL COM INTEGRAÇÃO ANALÍTICA

O cálculo das integrais que definem os esforços resistentes do concreto é feito subdividindo-se a área da poligonal comprimida em um conjunto de trapézios (fig. 4.3), determinados pelos lados l_i da poligonal e pelo eixo v (logicamente, o trapézio pode degenerar em retângulo ou triângulo). Assim, a solução da integração recai na integração das tensões em um trapézio genérico (fig. 4.4).

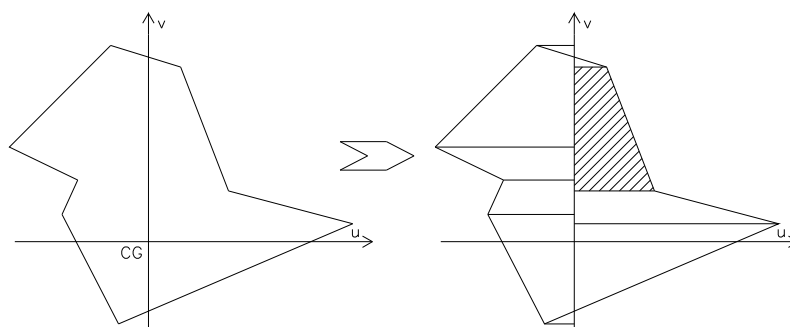


FIGURA 4.3 - Decomposição da poligonal em trapézios.

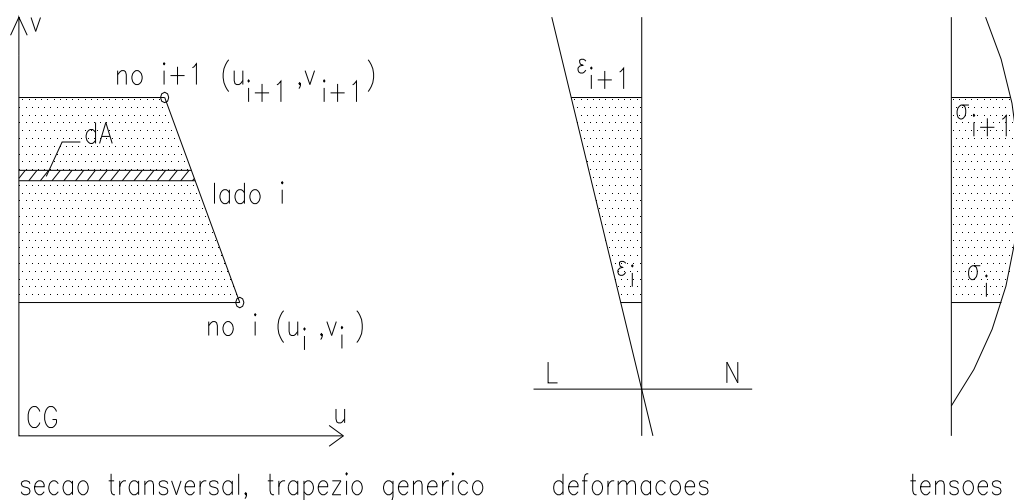


FIGURA 4.4 - Integração das tensões em um trapézio genérico.

FORMULAÇÃO PARA O CÁLCULO DOS ESFORÇOS RESISTENTES DE UM TRAPÉZIO GENÉRICO

a) ao longo do lado i , cálculo de $u=f(v)$: (ver fig. 4.4)

equação de reta $\rightarrow u=A.v+B$

em $u=u_i \rightarrow v=v_i \Rightarrow u_i=A.v_i+B$

em $u=u_{i+1} \rightarrow v=v_{i+1} \Rightarrow u_{i+1}=A \cdot v_{i+1}+B$

do sistema, tem-se:

$$u=A \cdot v+B \quad (4.6)$$

$$\text{com: } A = \frac{u_{i+1} - u_i}{v_{i+1} - v_i} \quad (4.7)$$

$$B = u_i - A \cdot v_i \quad (4.8)$$

b) cálculo de $v=f(\varepsilon)$:

isolando- se v na eq. 3.18, vem:

$$v = \frac{h_\alpha}{\theta} (\varepsilon - \varepsilon_{CG})$$

ou ainda:

$$v = \frac{h_\alpha}{\theta} \varepsilon - \frac{h_\alpha \cdot \varepsilon_{CG}}{\theta}$$

(4.9)

c) ao longo do lado i , cálculo de $u=f(\varepsilon)$: (ver fig. 4.4)

substituindo-se a eq. 4.9 na eq. 4.6, tem-se:

$$u = A \left(\frac{h_\alpha}{\theta} \varepsilon - \frac{h_\alpha \cdot \varepsilon_{CG}}{\theta} \right) + B$$

ou ainda:

$$u = \frac{A \cdot h_\alpha}{\theta} \varepsilon - \frac{A \cdot h_\alpha \cdot \varepsilon_{CG}}{\theta} + B$$

substituindo-se o valor de B dado na eq. 4.8, vem:

$$u = \frac{A \cdot h_\alpha}{\theta} \varepsilon - \frac{A \cdot h_\alpha \cdot \varepsilon_{CG}}{\theta} + u_i - A \cdot v_i$$

ou ainda:

$$u = \frac{A \cdot h_\alpha}{\theta} \varepsilon + u_i - A \left(v_i + \frac{h_\alpha \cdot \varepsilon_{CG}}{\theta} \right)$$

esta equação pode ser escrita como:

$$u = \frac{a \cdot h_\alpha}{\theta} \varepsilon + b \quad (4.10)$$

$$\text{com: } a = \frac{u_{i+1} - u_i}{v_{i+1} - v_i} \quad (4.11)$$

$$b = u_i - a \left(v_i + \frac{h_\alpha \cdot \varepsilon_{CG}}{\theta} \right) \quad (4.12)$$

d) definição do elemento infinitesimal - cálculo de $dA=f(\varepsilon)$: (ver fig. 4.4)

$$dA = u \cdot dV \quad (4.13)$$

mas, derivando-se a eq. 4.9 em relação a ε , vem:

$$\frac{dv}{d\varepsilon} = \frac{h_\alpha}{\theta}$$

ou ainda:

$$dv = \frac{h_\alpha}{\theta} d\varepsilon$$

que, substituído na eq. 4.13, fornece

$$dA = \frac{h_\alpha}{\theta} u \cdot d\varepsilon \quad (4.14)$$

e) cálculo de F_n :

para a simplificação das integrais que serão feitas, surge a necessidade da seguinte definição:

$$F_n = \left(\frac{h_\alpha}{\theta} \right)^{n+1} \int_{\varepsilon_i}^{\varepsilon_{i+1}} \varepsilon^n \sigma_c(\varepsilon) d\varepsilon \quad (4.15)$$

substituindo-se o valor dado pela eq. 3.36 na eq. 4.15, tem-se:

$$F_n = fc \left(\frac{h_\alpha}{\theta} \right)^{n+1} \int_{\varepsilon_i}^{\varepsilon_{i+1}} \sum_{k=0}^{k=m} \frac{C_k}{(1+\phi)^k} \varepsilon^n \varepsilon^k d\varepsilon$$

ou ainda:

$$F_n = fc \left(\frac{h_\alpha}{\theta} \right)^{n+1} \int_{\varepsilon_i}^{\varepsilon_{i+1}} \sum_{k=0}^{k=m} \frac{C_k}{(1+\phi)^k} \varepsilon^{k+n} d\varepsilon$$

efetuando-se a integral, vem:

$$F_n = fc \left(\frac{h_\alpha}{\theta} \right)^{n+1} \sum_{k=0}^{k=m} \frac{C_k}{(1+\phi)^k} \frac{\varepsilon^{k+n+1}}{k+n+1} \Big|_{\varepsilon_i}^{\varepsilon_{i+1}}$$

que resulta:

$$F_n = fc \left(\frac{h_\alpha}{\theta} \right)^{n+1} \sum_{k=0}^{k=m} \frac{C_k}{(1+\phi)^k} \frac{\varepsilon_{i+1}^{k+n+1} - \varepsilon_i^{k+n+1}}{k+n+1}$$

substituindo-se ε pelo valor dado na eq. 3.18, finalmente tem-se:

$$F_n = fc \left(\frac{h_\alpha}{\theta} \right)^{n+1} \sum_{k=0}^{k=m} \frac{C_k}{(1+\phi)^k} \frac{\left(\varepsilon_{CG} + \frac{\theta \cdot v_{i+1}}{h_\alpha} \right)^{k+n+1} - \left(\varepsilon_{CG} + \frac{\theta \cdot v_i}{h_\alpha} \right)^{k+n+1}}{k+n+1} \quad (4.16)$$

f) cálculo dos esforços resistentes fornecidos por um trapézio genérico: (ver fig. 4.4)

f.1) força normal:

estendendo-se a eq. 4.3 somente à área do trapézio, vem:

$$\Delta N_{rc} = \int_{l_i} \sigma_c \cdot dA$$

substituindo-se dA pelo valor dado na eq. 4.14, temos:

$$\Delta N_{rc} = \int_{l_i} \sigma_c \frac{h_\alpha}{\theta} u \cdot d\varepsilon = \frac{h_\alpha}{\theta} \int_{l_i} \sigma_c \cdot u \cdot d\varepsilon$$

substituindo-se u pelo valor dado pela eq. 4.10, vem:

$$\Delta N_{rc} = \frac{h_\alpha}{\theta} \int_{l_i} \sigma_c \left(\frac{a \cdot h_\alpha}{\theta} \varepsilon + b \right) d\varepsilon$$

ou ainda:

$$\Delta N_{rc} = b \left(\frac{h_\alpha}{\theta} \right)^1 \int_{l_i} \sigma_c \cdot d\varepsilon + a \left(\frac{h_\alpha}{\theta} \right)^2 \int_{l_i} \varepsilon \cdot \sigma_c \cdot d\varepsilon$$

comparando-se esta equação com a equação 4.15, vem que:

$$\Delta N_{rc} = b \cdot F_0 + a \cdot F_1 \quad (4.17)$$

com: a e $b \rightarrow$ eq. 4.11 e 4.12

$$F_0 \text{ e } F_1 \rightarrow \text{eq. 4.16}$$

f.2) momento fletor no sentido u :

estendendo-se a eq. 4.4 somente à área do trapézio, vem:

$$\Delta M_{ruc} = \int_{l_i} \sigma_c \cdot u \cdot dA$$

substituindo-se dA pelo valor dado na eq. 4.14, e sabendo-se que a abscissa do CG do elemento infinitesimal é $u/2$, temos:

$$\Delta M_{ruc} = \int_{l_i} \sigma_c \cdot \frac{u}{2} \frac{h_\alpha}{\theta} u \cdot d\varepsilon = \frac{h_\alpha}{2\theta} \int_{l_i} \sigma_c \cdot u^2 d\varepsilon$$

substituindo-se u pelo valor dado pela eq. 4.10, vem:

$$\Delta M_{ruc} = \frac{h_\alpha}{2\theta} \int_{l_i} \sigma_c \left(\frac{a \cdot h_\alpha}{\theta} \varepsilon + b \right)^2 d\varepsilon$$

ou ainda:

$$\Delta M_{ruc} = \frac{b^2}{2} \left(\frac{h_\alpha}{\theta} \right)^1 \int_{l_i} \sigma_c \cdot d\varepsilon + a \cdot b \left(\frac{h_\alpha}{\theta} \right)^2 \int_{l_i} \varepsilon \cdot \sigma_c d\varepsilon + \frac{a^2}{2} \left(\frac{h_\alpha}{\theta} \right)^3 \int_{l_i} \varepsilon^2 \cdot \sigma_c d\varepsilon$$

comparando-se esta equação com a equação 4.15, vem que:

$$\Delta M_{\text{rvc}} = \frac{b^2}{2} F_0 + a \cdot b \cdot F_1 + \frac{a^2}{2} F_2 \quad (4.18)$$

com: a e b \rightarrow eq. 4.11 e 4.12

F_0, F_1 e $F_2 \rightarrow$ eq. 4.16

f.3) momento fletor no sentido v:

estendendo-se a eq. 4.5 somente à área do trapézio, vem:

$$\Delta M_{\text{rvc}} = \int_{l_i} \sigma_c \cdot v \cdot dA$$

substituindo-se dA pelo valor dado na eq. 4.14, tem-se:

$$\Delta M_{\text{rvc}} = \int_{l_i} \sigma_c \cdot v \cdot \frac{h_\alpha}{\theta} u \cdot d\varepsilon = \frac{h_\alpha}{\theta} \int_{l_i} \sigma_c \cdot u \cdot v \cdot d\varepsilon$$

substituindo-se u pelo valor dado pela eq. 4.10 e v pelo valor dado pela eq. 4.9, vem:

$$\Delta M_{\text{rvc}} = \frac{h_\alpha}{\theta} \int_{l_i} \sigma_c \left(\frac{a \cdot h_\alpha}{\theta} \varepsilon + b \right) \left(\frac{h_\alpha}{\theta} \varepsilon - \frac{h_\alpha}{\theta} \varepsilon_{\text{CG}} \right) d\varepsilon$$

ou ainda:

$$\Delta M_{\text{rvc}} = b \left(\frac{h_\alpha}{\theta} \right)^2 \int_{l_i} \varepsilon \cdot \sigma_c d\varepsilon + a \left(\frac{h_\alpha}{\theta} \right)^3 \int_{l_i} \varepsilon^2 \cdot \sigma_c d\varepsilon - \frac{\varepsilon_{\text{CG}} \cdot h_\alpha}{\theta} \left[b \left(\frac{h_\alpha}{\theta} \right)^1 \int_{l_i} \sigma_c d\varepsilon + a \left(\frac{h_\alpha}{\theta} \right)^2 \int_{l_i} \varepsilon \cdot \sigma_c d\varepsilon \right]$$

comparando-se esta equação com as equações 4.15 e 4.17, vem que:

$$\Delta M_{\text{rvc}} = b \cdot F_1 + a \cdot F_2 - \frac{\varepsilon_{\text{CG}} \cdot h_\alpha}{\theta} \Delta N_{\text{rc}} \quad (4.19)$$

com: a e b \rightarrow eq. 4.11 e 4.12

F_1 e $F_2 \rightarrow$ eq. 4.16

$\Delta N_{\text{rc}} \rightarrow$ eq. 4.17

g) marcha de cálculo:

- 1) $i=1$ (i =lado que se está analisando);
- 2) se lado i for paralelo ao eixo u, vá para 12;
- 3) cálc. F_0, F_1 e $F_2 \rightarrow$ eq. 4.16;
- 4) cálc. a \rightarrow eq. 4.11;
- 5) cálc. b \rightarrow eq. 4.12;
- 6) cálc. $\Delta N_{\text{rc}} \rightarrow$ eq. 4.17;
- 7) cálc. $\Delta M_{\text{rvc}} \rightarrow$ eq. 4.18;
- 8) cálc. $\Delta M_{\text{rvc}} \rightarrow$ eq. 4.19;
- 9) $N_{\text{rc}} = N_{\text{rc}} + \Delta N_{\text{rc}}$;

$$10) M_{ruc} = M_{ruc} + \Delta M_{ruc};$$

$$11) M_{rvc} = M_{rvc} + \Delta M_{rvc};$$

$$12) i = i + 1;$$

13) se $i \leq N - 1$, voltar ao passo 2 (N é o número de nós da poligonal em questão).

h) observações:

h1) deve-se notar que o cálculo acima deve ser feito com a poligonal de concreto comprimida, e não com o polígono que define toda a seção; por área de concreto comprimida, subentende-se a área de concreto realmente sob compressão e eventuais áreas de concreto tracionado onde ainda age o diagrama σ - ε do concreto, quando este assim o prever; ainda, no caso de diagrama σ - ε do concreto ser formado por mais de uma curva polinomial, o cálculo deve ser feito separadamente, com poligonais de concreto onde age somente uma curva polinomial dentre as várias que formam o diagrama σ - ε do concreto, após o que se somam os esforços;

h2) como pode ser observado no item g acima, lado paralelo ao eixo u não contribui com esforços resistentes (área=0, ou seja, não há trapézio) e ocasionaria erro ao se calcular o valor da eq. 4.11;

h3) a formulação desenvolvida não é válida quando $\theta=0$ (veja as equações 4.12, 4.16 e 4.19), devendo ser utilizada uma formulação especialmente desenvolvida para este caso (tal formulação é deduzida no item 4.3.6);

h4) eventuais vazios são automaticamente levados em conta pelo método: como o sentido de circunscção dos nós dos vazios é invertido, obtém-se os valores das integrais referentes aos vazios com sinal trocado; ao se efetuar a soma destes valores, os vazios são automaticamente subtraídos;

h5) o efeito da fluência é levado em consideração, como pode ser observado na equação 4.16;

h6) caso se optasse pelo Processo da Poligonal com Integração Numérica, a única alteração na metodologia exposta neste item seria no cálculo dos valores F_n ; a equação 4.15 deveria ser resolvida através de um método numérico como, por exemplo, o Método da Quadratura de Gauss; em SANTOS (1994), é observada boa precisão utilizando-se 8 pontos de Gauss;

h7) a materialização computacional deste item é a sub-rotina "INTEGRAL_CONCRETO", constante do Apêndice 1.

4.3.4. CONSIDERAÇÕES SOBRE A PRECISÃO

Em SANTOS (1994), o autor observa a existência de imprecisões numéricas no cálculo dos esforços resistentes do concreto quando se utiliza o Processo da Poligonal com Integração Analítica, o que foi parcialmente resolvido com a implementação do Processo da Poligonal com Integração Numérica (Quadratura de Gauss). O autor do citado trabalho chegou à conclusão de que a imprecisão numérica aparece quando:

- a) a força normal solicitante é grande ($N_s \geq 0,8N_{r,MAXIMO}$): força normal alta obriga a um estado de deformação quase uniforme (valor baixo para θ - LN muito afastada da seção), o que causa imprecisões no cálculo de algumas fórmulas (veja que nas equações 4.12 e 4.19 θ aparece no divisor);
- b) a direção da LN é quase paralela a qualquer um dos lados da poligonal comprimida ($\Delta_{\text{ANG}} \leq 3$ graus): observe que a formulação deduzida não é válida para lado perfeitamente paralelo ao eixo u (veja as equações 4.11 e 4.16) que, aliás, não contribuem com esforços resistentes.

Cabe observar que o trabalho citado acima:

- a) utiliza fórmulas deduzidas exclusivamente para o diagrama σ - ε parábola-retângulo (NB1/78);
- b) utiliza o parâmetro β_v (profundidade adimensional da LN; $\beta_v = v_{LN}/h_\alpha$), enquanto o presente trabalho utiliza o parâmetro θ ;
- c) utiliza a linguagem de programação BASIC, enquanto o presente trabalho utiliza a linguagem FORTRAN.

No presente trabalho, após inúmeros testes utilizando-se a metodologia de cálculo deduzida no item 4.3.3 e implementada computacionalmente na linguagem FORTRAN, observou-se precisão praticamente exata. Executou-se, inclusive, o exemplo teste dado no item 8.4.1 de SANTOS (1994), não sendo encontrado nenhum tipo de imprecisão. Com a metodologia aqui exposta, pequenas imprecisões somente foram observadas quando:

- a) $\theta \leq 1.10^{-6}$ (observe que para esta curvatura, o estado de deformação é praticamente uniforme): como solução, quando se tem $\theta \leq 1.10^{-6}$, propõe-se o cálculo com $\theta = 1.10^{-6}$ e com $\theta = 0$ (sub-rotina especial para estado de deformação uniforme), interpolando-se os resultados;

- b) poligonal comprimida com lado quase paralelo à LN ($\Delta_{\text{ANG}} \leq 0,001$ grau): como solução quando se tem lado com esta característica, propõe-se desprezar a contribuição resistente de tal lado, já que esta contribuição seria realmente muito pequena.

4.3.5. OBTENÇÃO DA POLIGONAL COMPRIMIDA

Como observado no item 4.3.3-h, o cálculo da parcela dos esforços resistentes devida ao concreto deve ser feito não com a poligonal total que define a seção, mas separadamente com as poligonais onde agem cada uma das N_m curvas do diagrama σ - ε adotado para o concreto. Então, no cálculo da contribuição resistente do concreto, devem ser *montadas* N_m poligonais, e estas poligonais devem obedecer ao que determina o item 2.7 (numeração sequencial dos nós; sentido de circuição trigonométrico, com vazios no sentido contrário; etc.)¹⁴.

Seja, então, a curva i do diagrama σ - ε adotado para o concreto. Define-se a ordenada onde começa a agir a curva i por $V_{\text{corte}1_i}$, e a ordenada onde termina de agir a curva i por $V_{\text{corte}2_i}$. Assim, utilizando-se a eq. 3.18 e as definições do item 3.3.2, vem:

$$\begin{aligned} V_{\text{corte}1_i} &= \left\{ (1 + \phi) \cdot \text{MATRIZ_e_LIMITES}(i,1) - \varepsilon_{\text{CG}} \right\} \frac{h_{\alpha}}{\theta} \\ V_{\text{corte}2_i} &= \left\{ (1 + \phi) \cdot \text{MATRIZ_e_LIMITES}(i,2) - \varepsilon_{\text{CG}} \right\} \frac{h_{\alpha}}{\theta} \end{aligned} \quad (4.20)$$

¹⁴ A materialização computacional deste item é a sub-rotina “EXTRAI_POLIGONAL”, constante do Apêndice 1.

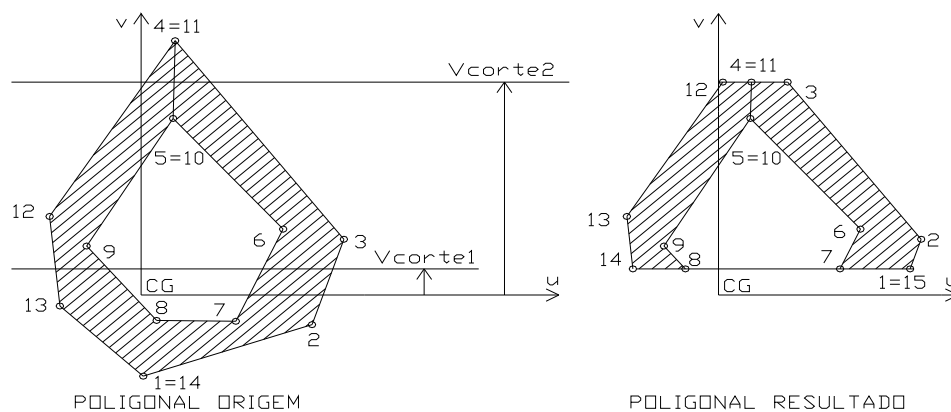


FIGURA 4.5 - Exemplo de poligonais origem e resultado.

Seja, então, uma poligonal “origem” (fig. 4.5): dada as ordenadas V_{corte1} e V_{corte2} , pede-se a poligonal “resultado”, que resulta da *extração* da parte da poligonal “origem” que se encontra entre as ordenadas V_{corte1} e V_{corte2} . A poligonal “resultado” deve obedecer às regras definidas no item 2.7, para que possa ser utilizada na metodologia de cálculo da contribuição resistente do concreto deduzida no item 4.3.3. Trata-se de um problema meramente geométrico, e, por isso, aqui não é discutido o embasamento teórico do método de resolução do problema. Limita-se apenas a se expor a marcha de cálculo:

- 1) $i=1$ (i é o número do nó que se está analisando da poligonal origem);
- 2) $k=0$ (k é o número do nó que se está analisando da poligonal resultado);
- 3) se o nó i estiver entre as ordenadas V_{corte1} e V_{corte2} (ver fig. 4.5):

$$k = k+1$$

$$u_{resultado}(k) = u_{origem}(i)$$

$$v_{resultado}(k) = v_{origem}(i)$$

- 4) se o lado i atravessar a ordenada V_{corte1} e não atravessar a ordenada V_{corte2} :

$$U_{corte} = u_{origem}(i) - (v_{origem}(i) - V_{corte1}) \frac{u_{origem}(i+1) - u_{origem}(i)}{v_{origem}(i+1) - v_{origem}(i)}$$

$$k = k+1$$

$$u_{resultado}(k) = U_{corte}$$

$$v_{resultado}(k) = V_{corte1}$$

- 5) se o lado i atravessar a ordenada V_{corte1} e não atravessar a ordenada V_{corte2} :

$$U_{corte} = u_{origem}(i) - (v_{origem}(i) - V_{corte2}) \frac{u_{origem}(i+1) - u_{origem}(i)}{v_{origem}(i+1) - v_{origem}(i)}$$

$$k = k+1$$

$$u_{resultado}(k) = U_{corte}$$

$$v_resultado(k) = Vcorte2$$

6) se o lado i atravessar as ordenadas $Vcorte1$ e $Vcorte2$:

6.1) se o lado i for ascendente ($v_origem(i+1) > v_origem(i)$):

$$Ucorte = u_origem(i) - (v_origem(i) - Vcorte1) \frac{u_origem(i+1) - u_origem(i)}{v_origem(i+1) - v_origem(i)}$$

$$k = k+1$$

$$u_resultado(k) = Ucorte$$

$$v_resultado(k) = Vcorte1$$

$$Ucorte = u_origem(i) - (v_origem(i) - Vcorte2) \frac{u_origem(i+1) - u_origem(i)}{v_origem(i+1) - v_origem(i)}$$

$$k = k+1$$

$$u_resultado(k) = Ucorte$$

$$v_resultado(k) = Vcorte2$$

6.2) se o lado i for descendente ($v_origem(i+1) < v_origem(i)$):

$$Ucorte = u_origem(i) - (v_origem(i) - Vcorte2) \frac{u_origem(i+1) - u_origem(i)}{v_origem(i+1) - v_origem(i)}$$

$$k = k+1$$

$$u_resultado(k) = Ucorte$$

$$v_resultado(k) = Vcorte2$$

$$Ucorte = u_origem(i) - (v_origem(i) - Vcorte1) \frac{u_origem(i+1) - u_origem(i)}{v_origem(i+1) - v_origem(i)}$$

$$k = k+1$$

$$u_resultado(k) = Ucorte$$

$$v_resultado(k) = Vcorte1$$

7) $i=i+1$

8) se $i \leq N-1 \rightarrow$ volta-se ao passo 3 (N é o número total de nós da poligonal origem)

9) colocação do nó de fechamento da poligonal resultado:

$$k = k+1$$

$$u_resultado(k) = u_resultado(1)$$

$$v_resultado(k) = v_resultado(1)$$

4.3.6. CÁLCULO DA CONTRIBUIÇÃO RESISTENTE DO CONCRETO NO CASO DE ESTADO DE DEFORMAÇÃO UNIFORME ($\theta=0$)

Como é explicado no item 4.3.3-h3, o cálculo da contribuição resistente do concreto quando se tem estado de deformação uniforme ($\theta=0$) deve ser feito através de uma formulação desenvolvida especialmente para este caso, pois $\theta=0$ causaria erro (divisão por zero) em algumas fórmulas do Processo da Poligonal com Integração Analítica.

Salienta-se que estado de deformação uniforme tão somente implica em $\theta=0$, não estando ε_{CG} obrigado a assumir um valor específico. Portanto, o estado de deformação da seção pode ou não obedecer ao ELU.

Quando se tem $\theta=0$, a tensão é constante em toda a seção transversal. Logo, o cálculo deve ser feito com toda a poligonal de concreto. As equações 4.3 a 4.5 se transformam, então, em:

$$N_{rc} = \sigma_c \cdot A_c \quad (4.21)$$

$$M_{rue} = N_{rc} \cdot U_{cg} \quad (4.22)$$

$$M_{rvc} = N_{rc} \cdot V_{cg} \quad (4.23)$$

onde: A_c : área da poligonal total que define a seção de concreto;

U_{cg} e V_{cg} : coordenadas(sist. Ouv) do CG da seção de concreto; caso tenha sido obedecida a orientação de colocar a origem do sistema de coordenadas no CG da seção, tem-se $U_{cg}=V_{cg}=0$;

σ_c : tensão correspondente à deformação ε_{CG} dada.

Seja, então, o cálculo de σ_c :

- em primeiro lugar, deve ser encontrada a curva do diagrama σ - ε do concreto que contém o ε_{CG} dado; algoritmo (veja as definições do item 3.3.2):

1) curva=0 (curva é o número do polinômio que se quer encontrar)

2) $i=1$

3) se $\varepsilon_{CG} \geq (1+\phi) \cdot \text{MATRIZ_e_LIMITES}(i,1)$ e $\varepsilon_{CG} \leq (1+\phi) \cdot \text{MATRIZ_e_LIMITES}(i,2)$:

curva=i

4) $i=i+1$

5) se $i \leq N_m$: volta-se ao passo 3

- 6) se curva=0: ϵ_{CG} dado não faz parte do diagrama σ - ϵ adotado para o concreto. $\therefore \sigma_c=0$
- com a curva do diagrama σ - ϵ encontrada como indicado acima, utiliza-se a eq. 3.36 para o cálculo de σ_c .

Passemos, agora, ao cálculo de A_c , U_{cg} e V_{cg} : conforme demonstrado em SANTOS(1981), estes parâmetros geométricos podem ser calculados através de:

$$aux_i = u_i \cdot v_{i+1} - u_{i+1} \cdot v_i$$

$$A_c = \frac{1}{2} \sum_{i=1}^{N-1} aux_i \quad (4.24)$$

$$S_{uu} = \frac{1}{6} \sum_{i=1}^{N-1} (v_i + v_{i+1}) aux_i \quad (4.25)$$

$$S_{vv} = \frac{1}{6} \sum_{i=1}^{N-1} (u_i + u_{i+1}) aux_i \quad (4.26)$$

$$U_{cg} = \frac{S_{vv}}{A_c} \quad (4.27)$$

$$V_{cg} = \frac{S_{uu}}{A_c} \quad (4.28)$$

onde: i : número do nó da poligonal total de concreto;

N : número total de nós da poligonal total de concreto;

S_{uu} : momento estático da seção em relação ao eixo u ;

S_{vv} : momento estático da seção em relação ao eixo v .

CAPÍTULO 5 - DETERMINAÇÃO DO ESTADO DE DEFORMAÇÃO DA SEÇÃO

5.1. INTRODUÇÃO

Este capítulo é de fundamental importância, uma vez que a determinação do estado de deformação da seção é um passo básico na análise da estabilidade de pilares.

Em um problema de análise não-linear física e geométrica de elementos lineares de concreto armado, o assunto deste capítulo é o responsável pela resolução da não-linearidade física. Como tal, pode ser utilizado tanto na análise da estabilidade de pilares isolados (que é o objeto de estudo do capítulo 7) quanto em análises não-lineares mais avançadas (pórtico tridimensional hiperestático, por exemplo).

No item 1.4 comenta-se que muitos algoritmos, supostos gerais e utilizados na grande maioria dos trabalhos correlatos a este, apresentam problemas de convergência com seções assimétricas sob determinadas condições de carregamento. Neste capítulo, explica-se a razão de tal anomalia e propõe-se algoritmos detalhados que pretendem ser eficazes com qualquer tipo de seção e sob quaisquer condições. Este é o capítulo mais importante deste trabalho.

Para o entendimento dos algoritmos que são apresentados neste capítulo, é fundamental o estudo prévio do Anexo A - Métodos numéricos para o cálculo de zeros de funções.

5.2. APRESENTAÇÃO DO PROBLEMA

Conforme vimos, o assunto do capítulo 4 trata do seguinte problema: dado o estado de deformação (θ , α e ε_{CG}) de uma seção definida geométrica¹⁵ e mecanicamente¹⁶, calcular os seus esforços resistentes (N_r , M_{rx} e M_{ry}). Esquemáticamente:

$$\left\{ \begin{array}{c} \text{seção} \\ \alpha \\ \theta \\ \varepsilon_{CG} \end{array} \right\} \Rightarrow \left\{ \begin{array}{c} N_r \\ M_{rx} \\ M_{ry} \end{array} \right\}$$

Trata-se, pois, de um problema de solução direta, onde não é necessário o cálculo iterativo, do tipo tentativa e erro.

O objeto de estudo deste capítulo é, de certa forma, o inverso do do capítulo 4, ou seja: dado os esforços solicitantes de uma seção definida geométrica e mecanicamente, calcular o estado de deformação ao qual a seção chega para estar em equilíbrio. Esquemáticamente:

$$\left\{ \begin{array}{c} \text{seção} \\ N_s \\ M_{sx} \\ M_{sy} \end{array} \right\} \Rightarrow \left\{ \begin{array}{c} \alpha \\ \theta \\ \varepsilon_{CG} \end{array} \right\}, \text{ tal que } \left\{ \begin{array}{c} N_r \\ M_{rx} \\ M_{ry} \end{array} \right\} = \left\{ \begin{array}{c} N_s \\ M_{sx} \\ M_{sy} \end{array} \right\}$$

Não se trata de um problema de solução direta, pois não se pode explicitar analiticamente o estado de deformação em função dos esforços resistentes da seção. Lança-se mão, então, de processos iterativos: arbitra-se um estado de deformação \rightarrow calculam-se os esforços resistentes (método exposto no capítulo 4) \rightarrow comparam-se com os esforços solicitantes; caso não sejam iguais, a menos de uma tolerância previamente estabelecida, arbitra-se um novo estado de deformação e repete-se o processo.

5.3. PROCESSOS ITERATIVOS PARA A DETERMINAÇÃO DO ESTADO DE DEFORMAÇÃO

Quanto a maneira de se arbitrar o estado de deformação, há 2 tipos principais de processos iterativos:

- 1) PROCESSO ITERATIVO DIRETO
- 2) PROCESSO ITERATIVO INDIRETO

¹⁵ Poligonal que define a seção de concreto, posição e área das barras da armadura.

No 1º processo, são arbitrados θ , α e ε_{CG} diferentes em cada iteração. Veja, por exemplo, CABRÉ (1972), FORNI (1980), MARCOTTI (1984) e CAMPOS (1994) Esquemáticamente:

- 1) arbitra-se θ , α e ε_{CG}
- 2) calcula-se N_r , M_{rx} e M_{ry} (método exposto no capítulo 4)
- 3) se $N_r \cong N_s$, $M_{rx} \cong M_{sx}$ e $M_{ry} \cong M_{sy}$: ok; senão, volta-se ao passo 1

Para se aumentar a eficiência do processo, no passo 1 não se deve arbitrar θ , α e ε_{CG} ao acaso, mas utilizar-se de uma metodologia apropriada. O problema deste tipo de processo é justamente encontrar uma metodologia que seja ao mesmo tempo eficiente¹⁷ e eficaz¹⁸. Na bibliografia pesquisada até o momento, não se encontrou uma metodologia que atenda a estes dois requisitos (principalmente ao da eficácia), veja CABRÉ (1972) e FORNI (1980); por isso, o processo iterativo direto não é o utilizado neste trabalho.

No 2º processo, veja, por exemplo, SANTOS (1981), MARCOTTI (1984), MENDES NETO (1991), SANTOS (1994) e CAMPOS (1994), há um ramo iterativo para cada uma das três variáveis (θ , α e ε_{CG}); portanto, o processo iterativo indireto é composto por três processos iterativos independentes. Em cada um dos três processos, é arbitrada somente uma variável, o que facilita em muito a implementação de uma metodologia eficiente de arbitragem dessas variáveis. Esquemáticamente, tem-se:

- 1) arbitra-se θ
- 2) arbitra-se α
- 3) arbitra-se ε_{CG}
- 4) calcula-se N_r , M_{rx} e M_{ry} (método exposto no capítulo 4)
- 5) se $N_r \neq N_s \rightarrow$ volta-se ao passo 3
- 6) se $\beta_r \neq \beta_s \rightarrow$ volta-se ao passo 2 ($\beta = \arctan \frac{M_y}{M_x}$)
- 7) se $|M_r| \neq |M_s| \rightarrow$ volta-se ao passo 1 ($|M| = \sqrt{M_x^2 + M_y^2}$)

ou, dando nome a cada um dos processos iterativos independentes, tem-se a figura 5.1.

¹⁶ Equações constitutivas do concreto e do aço.

¹⁷ Minimização do número de iterações (menor tempo de processamento computacional).

¹⁸ Encontrar sempre a solução do problema (garantir a convergência do processo iterativo).

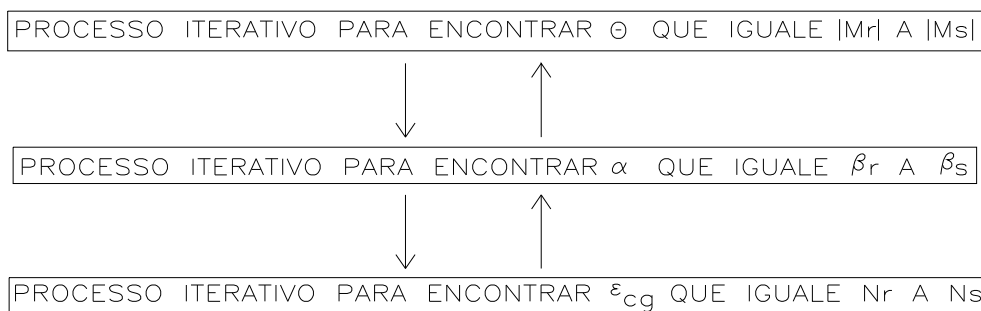


FIGURA 5.1 - Esquema geral do processo iterativo para a determinação do estado de deformação da seção.

Nos itens seguintes, cada um destes três processos iterativos são discutidos minuciosamente.

5.4. PROCESSO ITERATIVO PARA ENCONTRAR ϵ_{CG} QUE IGUALE N_r A N_s

O problema a ser resolvido neste item é: dada a seção totalmente definida (sist. Oxy), a curvatura majorada adimensional (θ), a direção da linha neutra (α) e a força normal solicitante (N_s), calcular a deformação do cg da seção (ϵ_{CG}) que faz com que a força normal resistente (N_r) se iguale a força normal solicitante dada¹⁹ (N_s). Esquemáticamente:

$$\left\{ \begin{array}{l} \text{seção} \\ \theta \\ \alpha \\ N_s \end{array} \right\} \Rightarrow \{ \epsilon_{CG}, \text{tal que } N_r = N_s \}$$

Observe que a resposta do problema não precisa, necessariamente, obedecer às restrições de norma quanto aos limites para ϵ_{CG} , podendo, portanto, o ϵ_{CG} calculado levar o estado de deformação a ultrapassar o ELU de ruptura do concreto comprimido ou de deformação plástica excessiva da armadura tracionada.

A resposta do problema exige um cálculo por tentativas:

- 1) arbitra-se ϵ_{CG}
- 2) calcula-se N_r , M_{rx} e M_{ry} → método exposto no capítulo 4
- 3) se $N_r \neq N_s$ → volta-se ao passo 1

Para otimizar este processo, deve-se lançar mão de um método numérico iterativo de cálculo de zeros de funções²⁰. A função, no caso, é $y=N_r(\epsilon_{CG})-N_s$. O comportamento típico da curva $N_r=f(\epsilon_{CG})$ está exemplificado na figura 5.3. Seleccionadamente, são três os métodos que podem ser utilizados: o Método da Secante, o Método da Bissecção e o Método da Posição Falsa.

¹⁹ A materialização computacional deste item é a sub-rotina “PROCURA_e_CG”, constante do Apêndice 1.

²⁰ Ver Anexo A ou RUGGIERO & LOPES(1988)

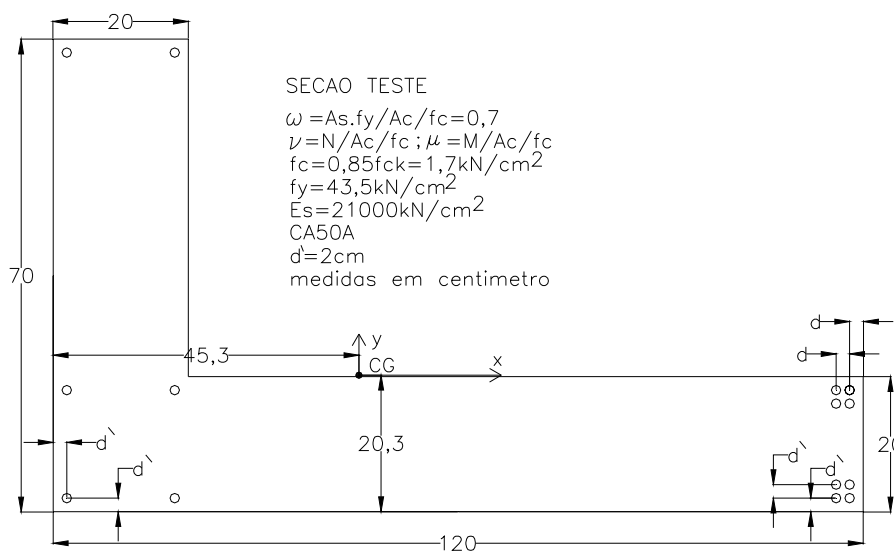
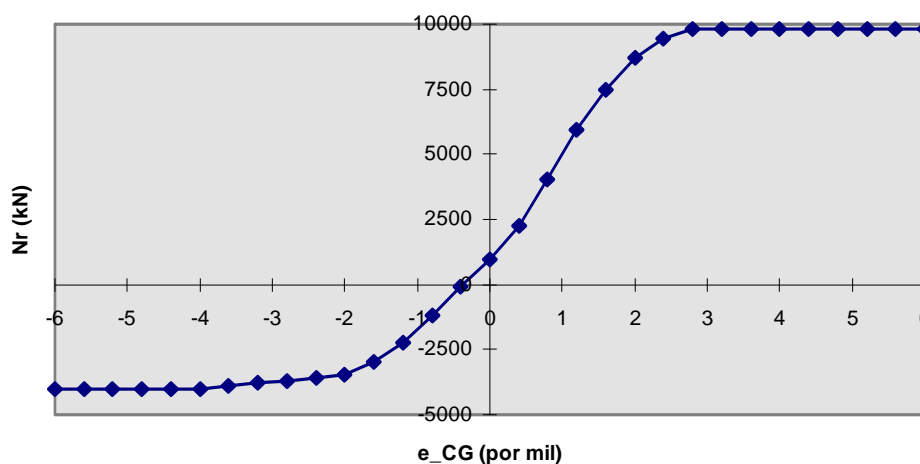


FIGURA 5.2 - Seção transversal teste.

seção teste(fig.5.2); teta=3; alfa=0

FIGURA 5.3 - Andamento típico da função $N_r = f(\epsilon_{CG})$.

Como a função $N_r = f(\epsilon_{CG})$ apresenta mudança de sinal na sua derivada segunda (veja a figura 5.3), o Método da Secante pode, dependendo da aproximação inicial adotada, jamais encontrar a solução do problema (divergir). Portanto, qualquer algoritmo que utilize o Método da Secante para otimizar o processo iterativo para encontrar ϵ_{CG} (ou a profundidade

adimensional da linha neutra, se for o caso), pode, sob determinadas condições, não funcionar²¹.

O Método da Bissecção tem as mesmas características do Método da Posição Falsa; porém, de um modo geral, este tem uma convergência mais rápida que aquele, sendo, então, preferível utilizar o Método da Posição Falsa ao Método da Bissecção.

Analisando-se o andamento da função $N_r=f(\varepsilon_{CG})$, figura 5.3, conclui-se que o Método da Posição Falsa é o mais adequado para o cálculo do zero de $y=N_r(\varepsilon_{CG})-N_s$, pois ele é eficiente e, principalmente, tem convergência garantida para o tipo de função em questão, ao contrário de outros métodos mais eficientes, como o da Secante.

Conforme explica-se no Anexo A, o Método da Posição Falsa necessita, para começar o processo iterativo, que se determine um intervalo inicial $[a,b]$ que contenha internamente a raiz procurada ($a \leq \varepsilon_{CG, \text{solução}} \leq b$). Como aproximação inicial para a e b , pode-se utilizar o $\varepsilon_{CG, \text{MÍNIMO}}$ e o $\varepsilon_{CG, \text{MÁXIMO}}$ que fazem com que o estado de deformação da seção atinja o ELU (ver item 3.2.2, eq. 3.26 e 3.27). Caso o intervalo assim determinado ainda não contenha em seu interior a raiz procurada, pode-se lançar mão do Método dos Intervalos Encaixantes para deslocar este intervalo até que este contenha a raiz.

A seguir, está o algoritmo proposto para o processo iterativo para encontrar ε_{CG} que iguale N_r a N_s :

- 1) definições iniciais: precisão $x=0,0000001\%$ e precisão $y=0,01\%$ (estes valores de precisão se mostraram ideais no decorrer de testes numéricos)
- 2) rotação do sistema de coordenadas Oxy para o Ouv (LN paralela ao eixo u): eq.4.1
- 3) determinação do intervalo inicial $[a,b]$:
 - 3.1) cálc. de $\varepsilon_{CG, \text{MÍNIMO}}$ e $\varepsilon_{CG, \text{MÁXIMO}}$: eq. 3.26 e 3.27
 - 3.2) cálc. de $N_{r, \text{min}}$ e $N_{r, \text{max}}$: método exposto no capítulo 4
 - 3.3) se $N_s > N_{r, \text{max}}$: Método dos Intervalos Encaixantes
 - 3.3.1) $\varepsilon_{CG, \text{MÍNIMO}} = \varepsilon_{CG, \text{MÁXIMO}}$
 - 3.3.2) $\varepsilon_{CG, \text{MÁXIMO}} = \varepsilon_{CG, \text{MÁXIMO}} + \text{passo}$ (passo=0,5%)
 - 3.3.3) $\varepsilon_{CG} = \varepsilon_{CG, \text{MÁXIMO}} \rightarrow$ cálc. $N_{r, \text{max}}$: método exposto no capítulo 4

²¹ Este é um dos motivos de alguns algoritmos, supostos gerais, não funcionarem em determinados

- 3.3.4) se $N_s > N_{r,max}$: volta-se ao passo 3.3.1
- 3.4) se $N_s < N_{r,min}$: Método dos Intervalos Encaixantes
- 3.4.1) $\varepsilon_{CG,MÁXIMO} = \varepsilon_{CG,MÍNIMO}$
- 3.4.2) $\varepsilon_{CG,MÍNIMO} = \varepsilon_{CG,MÍNIMO} + \text{passo}$ (passo=0,5%)
- 3.4.3) $\varepsilon_{CG} = \varepsilon_{CG,MÍNIMO} \rightarrow$ cálc. $N_{r,min}$: método exposto no capítulo 4
- 3.4.4) se $N_s < N_{r,min}$: volta-se ao passo 3.4.1
- 3.5) intervalo inicial: $a = \varepsilon_{CG,MÍNIMO}$; $ya = N_{r,min} - N_s$; $b = \varepsilon_{CG,MÁXIMO}$; $yb = N_{r,max} - N_s$
- 4) Cálculo da raiz: Método da Posição Falsa
- 4.1) $M = ya$; $yx = ya$
- 4.2) $x = (a \cdot yb - b \cdot ya) / (yb - ya)$
- 4.3) $\varepsilon_{CG} = x \rightarrow$ cálc. N_r , M_{rx} , M_{ry} : método exposto no capítulo 4; $yx = N_r - N_s$
- 4.4) se $M \cdot yx \geq 0$: $a = x$; $ya = yx$
- 4.5) se $M \cdot yx < 0$: $b = x$; $yb = yx$
- 4.6) se $|b - a| \leq \text{precisão } x$:
- 4.6.1) $\varepsilon_{CG} = (a + b) / 2 \rightarrow$ cálc. N_r , M_{rx} , M_{ry} : método exposto no capítulo 4
- 4.6.2) cálc. M_{rx} e M_{ry} : eq. 4.2
- 4.6.3) fim do processo
- 4.7) se $yx > \text{precisão } y \cdot N_s$: volta-se ao passo 4.2
- 4.8) cálc. M_{rx} e M_{ry} : eq. 4.2
- 4.9) fim do processo

5.5. PROCESSO ITERATIVO PARA ENCONTRAR α QUE IGUALE β_r A β_s

O problema a ser resolvido neste item é: dada a seção totalmente definida, a curvatura majorada adimensional (θ) e a direção do momento solicitante (β_s), calcular a direção da linha neutra (α) que faz com que a direção do momento resistente (β_r) seja a mesma do momento solicitante dado²². Logicamente, a condição $N_r = N_s$ também deve ser atendida, mas esta é encargo do item 5.4. Esquematicamente, tem-se:

$$\left\{ \begin{array}{l} \text{seção} \\ \theta \\ \beta_s \end{array} \right\} \Rightarrow \{ \alpha, \text{tal que } \beta_r = \beta_s \}$$

problemas, conforme comenta-se no item 1.2.

²² A materialização computacional deste item é a sub-rotina “PROCURA_ALFA”, constante do Apêndice 1.

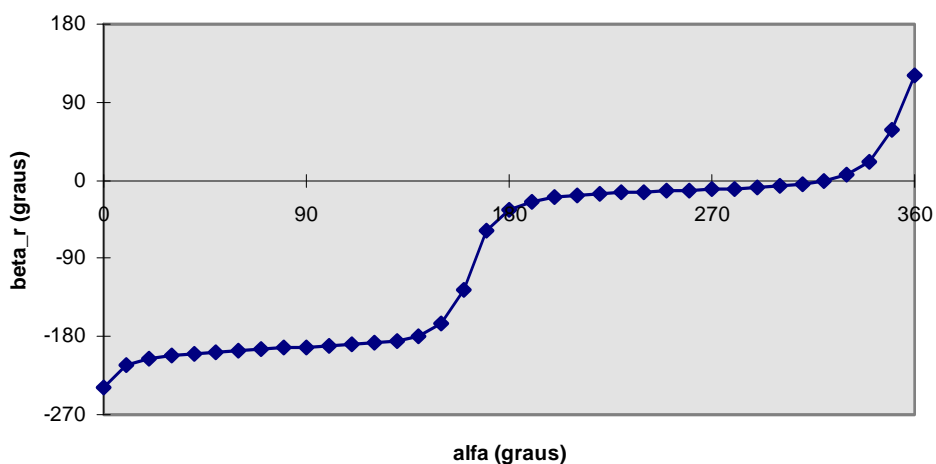
A resposta do problema exige um cálculo por tentativas:

- 1) arbitra-se α
- 2) calcula-se $\beta_r = \arctan(M_{ry}/M_{rx}) \rightarrow$ processo iterativo para encontrar ϵ_{CG} tal que $N_r = N_s$,
item 5.4
- 3) se $\beta_r \neq \beta_s \rightarrow$ volta-se ao passo 1

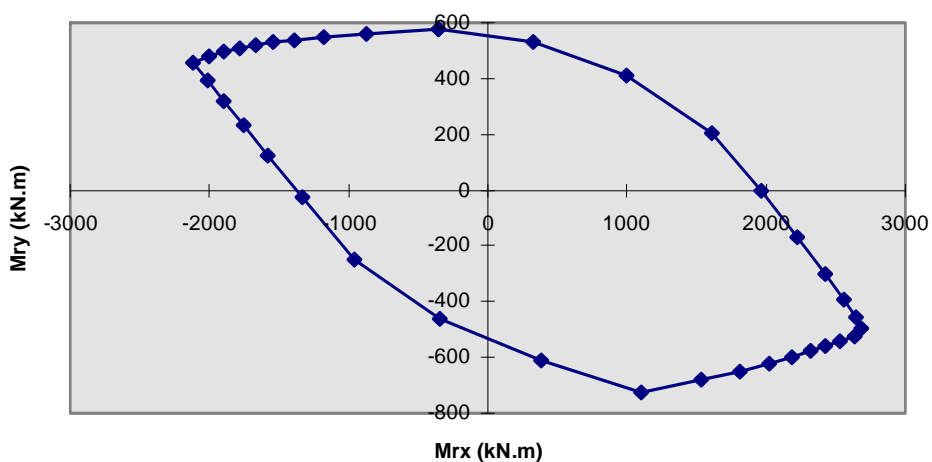
Porém, o problema não é tão simples quanto possa parecer: dependendo do tipo de seção transversal e do θ arbitrado, o problema pode ter 1, 2 ou nenhuma solução (α que possibilite $\beta_r = \beta_s$).

A função da qual se necessita o zero é $y = \beta_r(\alpha) - \beta_s$, interessando, pois, estudar o comportamento da curva $\beta_r = f(\alpha)$. A forma de tal curva, na grande maioria dos casos, está exemplificada na figura 5.4. Observa-se que, ao se variar α de 0 a 360° , a faixa de variação de β_r também é de 360° , sempre havendo um e somente um α para cada β_r . Observa-se, também, a existência de trechos de grande inclinação seguidos por trechos muito pouco inclinados, o que indica que métodos do tipo dos Intervalos Encaixantes precisam utilizar passo pequeno. Outra forma de representar o mesmo fenômeno está na figura 5.5, onde observa-se que a origem do sistema de coordenadas é concêntrica à curva e que esta é fechada.

seção teste(fig.5.2); teta=3; Nr=4046kN(ni=0,7)

FIGURA 5.4 - Andamento típico da função $\beta_r=f(\alpha)$.

seção teste(fig.5.2); teta=3; Nr=4046kN(ni=0,7)

FIGURA 5.5 - Andamento típico da curva $M_{rx}-M_{ry}$.

Porém, em problemas com determinadas condições, o comportamento da curva $\beta_r=f(\alpha)$ não é o descrito acima, daí a característica de alguns problemas terem 2 ou nenhuma solução (α que possibilite $\beta_r=\beta_s$). Explica-se: se a seção for assimétrica e a curvatura (θ) arbitrada for pequena, pode ocorrer que β_r se restrinja a uma pequena faixa de variação, mesmo que se arbitre α ao longo de todos os 360° (ver fig. 5.6 e 5.7). Se o β_s dado não

estiver dentro desta faixa, não haverá α que possibilite igualar β_r a β_s . Um exemplo no limite explica melhor: suponha uma seção assimétrica submetida a uma curvatura tão pequena que tende a zero (estado uniforme de deformação); neste caso, só há uma direção para o momento resistente, qualquer que seja a direção arbitrada para a linha neutra (a faixa de variação de β_r se restringe a um ponto!); logo, para este caso, não se pode impor um β_s qualquer, pois não há solução matemática²³. Quando se tem seção duplamente simétrica, não há problema, pois variando-se α de 0 a 360°, a faixa de variação de β_r também é de 360°, e sempre há solução (e única), conforme mostra a figura 5.4. O mesmo ocorre quando se tem seção assimétrica desde que a curvatura arbitrada não seja pequena ($|\theta| \geq 0,5$ por exemplo).

²³ Observe que isto não quer dizer, então, que a seção vai romper, mas simplesmente que para o θ arbitrado não há α que possibilite igualar β_r a β_s ; então, deve-se arbitrar outro θ e tentar novamente.

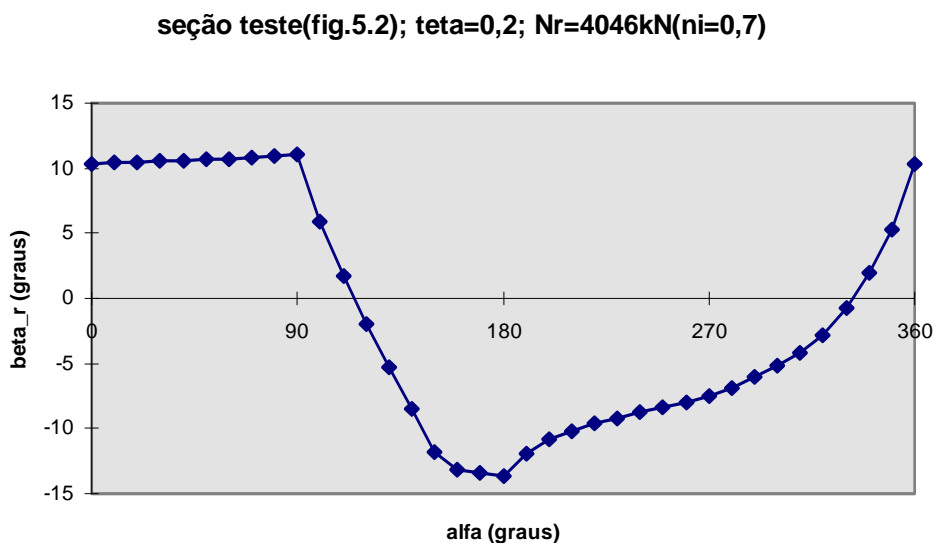


FIGURA 5.6 - Andamento da função $\beta_r=f(\alpha)$ para seção assimétrica e θ baixo.

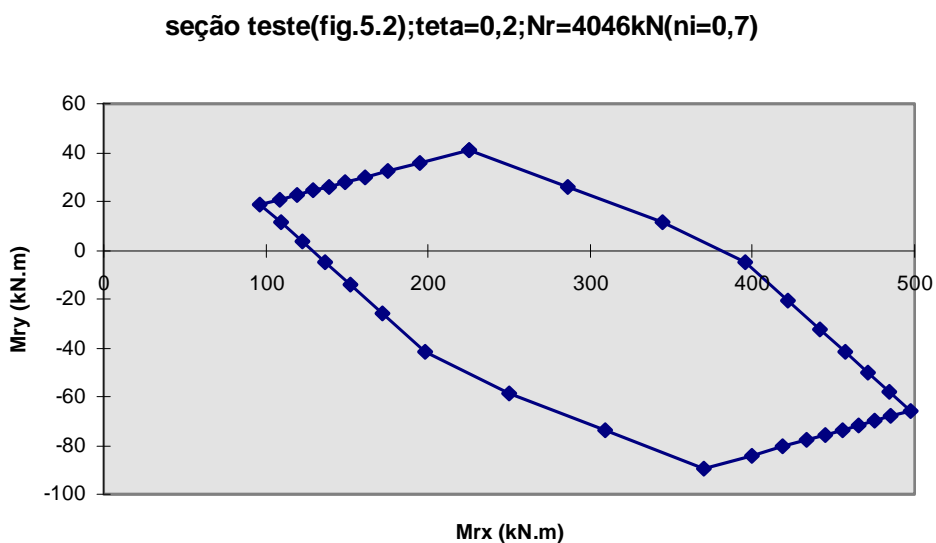


FIGURA 5.7 - Andamento da curva $M_{rx}-M_{ry}$ para seção assimétrica e θ baixo.

Ainda considerando o caso de β_r se restringir a uma pequena faixa de variação mesmo que se arbitre α ao longo de todos os 360° , se a direção dada para o momento solicitante estiver dentro desta faixa, existirão 2 direções para a linha neutra que possibilitam

igualar β_r a β_s (ver fig. 5.6 e 5.7)²⁴. Surge, então, mais um problema: qual das duas direções para a linha neutra é a resposta procurada? concluiu-se que a direção procurada é:

a) se $\theta \geq 0 \rightarrow$ o α procurado é aquele de maior $|M_r|$;

b) se $\theta < 0 \rightarrow$ o α procurado é aquele de menor $|M_r|$ ($|M_r| = \sqrt{M_{rx}^2 + M_{ry}^2}$);

tal conclusão não pode ser explicada somente com os dados deste item, mas analisando-se a forma da curva $|M_r|=f(\theta)$, curva 1 da fig. 5.9: compare a magnitude de $|M_r|$ nos dois trechos da curva (θ positivo e negativo). Porém, durante os testes numéricos, observou-se que não é preciso calcular as duas raízes do problema para somente após isso escolher uma delas, o que diminui o tempo de processamento computacional. A solução encontrada está detalhada no algoritmo a seguir.

No item 1.4 comenta-se que muitos algoritmos, supostos gerais e utilizados em muitos trabalhos, podem apresentar problemas e não funcionarem em problemas com seções assimétricas sob determinadas condições de carregamento (geralmente, força normal solicitante alta). Isso ocorre porque tais algoritmos não prevêm os casos explicados acima. Se, durante a execução computacional de tais algoritmos, for arbitrada uma curvatura (θ) para a qual não há nenhuma direção da linha neutra (α) que possibilite igualar a direção do momento resistente a do solicitante ($\beta_r=\beta_s$), o processo iterativo pode divergir; se, para a curvatura arbitrada, houverem duas direções da linha neutra que atendem a condição $\beta_r=\beta_s$, o processo iterativo pode fornecer a direção incorreta como resposta.

Quanto aos métodos numéricos que podem ser utilizados para otimizar o processo iterativo, observa-se que métodos do tipo Newton-Raphson ou Secante podem jamais encontrar a raiz (podem divergir). Isto ocorre porque, como pode ser observado na fig. 5.6, a função $\beta_r=f(\alpha)$ pode apresentar mudanças no sinal da sua derivada primeira (ver Anexo A).

O Método da Bissecção tem as mesmas características do Método da Posição Falsa; porém, de um modo geral, este tem uma convergência mais rápida que aquele, sendo, então, preferível utilizar o Método da Posição Falsa ao Método da Bissecção.

²⁴ Poder-se-ia indagar se não existem casos especiais onde há mais de duas direções para a linha neutra que atendem a esta condição. Após um grande número de testes numéricos com várias seções e condições, não se encontrou nenhum caso onde isso tenha ocorrido. A bibliografia internacional consultada chega a mesma conclusão, veja YAU et al. (1993).

Analisando-se o andamento da função $\beta_r=f(\alpha)$, fig. 5.4 e 5.6, conclui-se que o Método da Posição Falsa é o mais adequado para o cálculo do zero de $y=\beta_r(\alpha)-\beta_s$, pois ele é eficiente e, principalmente, tem convergência garantida para o tipo de função em questão, ao contrário de outros métodos mais eficientes, como o da Secante.

Conforme explica-se no Anexo A, o Método da Posição Falsa necessita, para começar o processo iterativo, que se determine um intervalo inicial $[a,b]$ que contenha internamente a raiz procurada ($a \leq \varepsilon_{CG, \text{solução}} \leq b$). Para se encontrar esse intervalo, pode-se lançar mão do Método dos Intervalos Encaixantes.

Esquemáticamente, então, o algoritmo do processo iterativo para encontrar α que iguale β_r a β_s fica:

- 1) definições iniciais: precisão x =precisão y =0,0001rad (0,0057graus); estes valores de precisão se mostraram ideais no decorrer de testes numéricos
- 2) EXISTE='sim' (EXISTE assume 'sim' ou 'não' caso exista ou não um α que possibilite igualar β_r a β_s)
- 3) se $\theta=0$: (rotina especial para estado uniforme de deformação)
 - 3.1) $\alpha=0$
 - 3.2) cálc. $\beta_r=\arctan(M_{ry}/M_{rx})$: processo iterativo para encontrar ε_{CG} tal que $N_r=N_s$, item 5.4
 - 3.3) se $\beta_r \neq \beta_s \rightarrow$ EXISTE='não'
 - 3.4) fim do processo iterativo
- 4) cálc. do 'a' inicial: como tentativa inicial, coloca-se a LN na direção perpendicular à do momento solicitante
 - 4.1) $a=\beta_s-\pi/2$
 - 4.2) cálc. y_a :
 - 4.2.1) $\alpha=a \rightarrow$ cálc. $y_a=\beta_r-\beta_s$, com $\beta_r=\arctan(M_{ry}/M_{rx})$: processo iterativo para encontrar ε_{CG} que iguale N_r a N_s , item 5.4
 - 4.2.2) se $y_a < -\pi \rightarrow y_a=y_a+2\pi$
 - 4.2.3) se $y_a > +\pi \rightarrow y_a=y_a-2\pi$
 - 4.3) se $|y_a| \leq$ precisão $y \rightarrow$ fim do processo iterativo
- 5) preparativos para o cálculo do 'b' inicial:
 - 5.1) determinação se α deve aumentar (variar no sentido trigonométrico) ou diminuir para poder encontrar o α_{RAIZ} procurado:

- 5.1.1) se $\text{sen}(\beta_r - \beta_s) < 0 \rightarrow \text{sinal} = +1$ (α deve aumentar)
- 5.1.2) se $\text{sen}(\beta_r - \beta_s) > 0 \rightarrow \text{sinal} = -1$ (α deve diminuir)
- 5.1.3) se $\theta < 0 \rightarrow \text{sinal} = -\text{sinal}$
- 5.2) determinação do passo a ser utilizado:
- 5.2.1) se a seção for duplamente simétrica $\rightarrow \text{passo} = \text{sinal} \cdot \pi / 12$ (15 graus)
- 5.2.1) se a seção não for duplamente simétrica $\rightarrow \text{passo} = \text{sinal} \cdot \pi / 36$ (5 graus)
- 5.3) no início do processo, 'b' deve ser igual a 'a': $b = a$; $y_b = y_a$
- 5.4) ao longo do processo, 'a' inicial será modificado: guardar o cálculo de 'a' feito com $\alpha = \beta_s - \pi / 2$: $a_0 = a$; $y_{a0} = y_a$
- 6) cálc. do intervalo inicial [a,b]: (Método dos Intervalos Encaixantes)
- 6.1) $a = b$; $y_a = y_b$
- 6.2) $b = b + \text{passo}$
- 6.3) se $|b - a| > \pi$: α já foi incrementado em 180° e não foi encontrada raiz \therefore não há raiz $\rightarrow \text{EXISTE} = \text{'não'}$; fim do processo iterativo
- 6.4) cálc. y_b :
- 6.4.1) $\alpha = b \rightarrow$ cálc. $y_b = \beta_r - \beta_s$, com $\beta_r = \arctan(M_{ry} / M_{rx})$: processo iterativo para encontrar ε_{CG} que iguale N_r a N_s , item 5.4
- 6.4.2) se $y_b < -\pi \rightarrow y_b = y_b + 2\pi$
- 6.4.3) se $y_b > +\pi \rightarrow y_b = y_b - 2\pi$
- 6.5) se $y_a \cdot y_b > 0 \rightarrow$ volta-se ao passo 6.1
- 7) cálculo da raiz: Método da Posição Falsa
- 7.1) $M = y_a$; $y_x = y_a$
- 7.2) se $M \cdot y_x \geq 0$: $a = x$; $y_a = y_x$
- 7.3) se $M \cdot y_x < 0$: $b = x$; $y_b = y_x$
- 7.4) se $|b - a| \leq \text{precisão } x$:
- 7.4.1) $\alpha = (a + b) / 2 \rightarrow$ cálc. N_r , M_{ru} , M_{rv} : processo iterativo para encontrar ε_{CG} que iguale N_r a N_s , item 5.4
- 7.4.2) fim do processo
- 7.6) $x = (a \cdot y_b - b \cdot y_a) / (y_b - y_a)$
- 7.7) cálc. y_x :
- 7.7.1) $\alpha = x \rightarrow$ cálc. $y_x = \beta_r - \beta_s$, com $\beta_r = \arctan(M_{ry} / M_{rx})$: processo iterativo para encontrar ε_{CG} que iguale N_r a N_s , item 5.4
- 7.7.2) se $y_x < -\pi \rightarrow y_x = y_x + 2\pi$
- 7.7.3) se $y_x > +\pi \rightarrow y_x = y_x - 2\pi$

7.8) se $y > \text{precisão}$ $y \rightarrow$ volta-se ao passo 7.2

7.9) fim do processo

5.6. PROCESSO ITERATIVO PARA ENCONTRAR θ QUE IGUALA $|M_r|$ A $|M_s|$

O problema a ser resolvido neste item é: dada a seção totalmente definida e o módulo do momento solicitante ($|M_s|$), calcular a curvatura majorada adimensional (θ) que propicia igualar o módulo do momento resistente ($|M_r|$) ao módulo do momento solicitante dado²⁵. Logicamente, as condições $N_r=N_s$ e $\beta_r=\beta_s$ também devem ser atendidas, mas isto é automaticamente garantido com o uso dos processos iterativos dados nos itens 5.4 e 5.5. Além disso, deve ser verificado se, para os esforços solicitantes dados, o estado de deformação necessário ao equilíbrio ultrapassa ou não o ELU de ruptura. Esquemáticamente, tem-se:

$$\left\{ \begin{array}{l} \text{secao} \\ |M_s| \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} \theta, \text{ tal que } |M_r| = |M_s| \\ \text{informacao se a secao rompeu ou nao} \end{array} \right\}$$

A resposta do problema exige um cálculo por tentativas:

1) arbitra-se θ

2) calcula-se $|M_r| = \sqrt{M_{rx}^2 + M_{ry}^2} \rightarrow$ processo iterativo para encontrar α tal que $\beta_r=\beta_s$,
item 5.5

3) se $|M_r| \neq |M_s| \rightarrow$ volta-se ao passo 1

4) verifica-se se o estado de deformação calculado ultrapassa ou não o ELU de ruptura \rightarrow
item 3.2.2, eq. 3.26 e 3.27

Para otimizar este processo, deve-se lançar mão de um método numérico iterativo de cálculo de zeros de funções²⁶. A função, no caso, é $y = |M_r|(\theta) - |M_s|$, sendo necessário, então, analisar o comportamento da curva $|M_r| = f(\theta)$, conhecida como diagrama momento fletor-força normal-curvatura (diagrama M-N- θ).

²⁵ A materialização computacional deste item é a sub-rotina “RELACAO_M_N_TETA”, constante do Apêndice 1.

²⁶ Ver Anexo A ou RUGGIERO & LOPES(1988)

Para seções duplamente simétricas, o diagrama M-N- θ tem a forma exemplificada na figura 5.8: observa-se que a curva é convexa, contínua e que passa pela origem (para curvatura igual a zero, o momento resistente da seção é nulo).

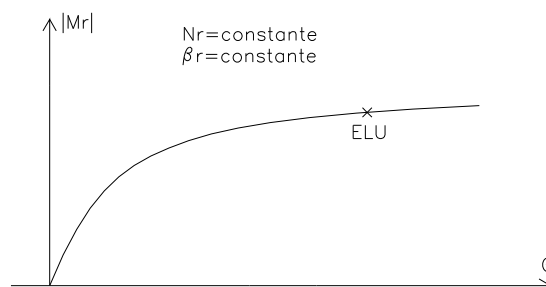


FIGURA 5.8 - Andamento típico do diagrama M-N- θ para seção simétrica.

Quando se tem seção assimétrica, no entanto, o andamento do diagrama M-N- θ pode ser muito diferente do da seção simétrica, conforme mostra a figura 5.9. O caso mais geral de diagrama M-N- θ é o representado pela curva 1 da figura 5.9²⁷: curva descontínua que não passa pela origem, com um trecho situado no ramo positivo do eixo θ e outro no negativo. Dependendo da direção imposta ao momento resistente (β_r), o diagrama M-N- θ pode, também, ser contínuo e estar situado todo no ramo positivo do eixo θ , conforme exemplifica a curva 2 da figura 5.9.

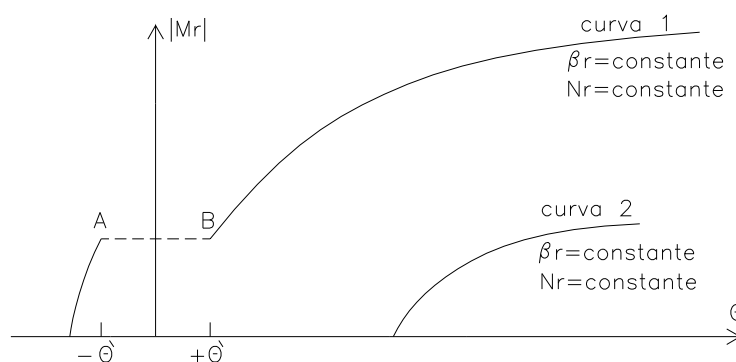


FIGURA 5.9 - Andamentos típicos do diagrama M-N- θ para seção assimétrica.

Após um grande número de testes numéricos com várias seções transversais e condições de carregamento (N_r e β_r), algumas observações podem ser feitas sobre o andamento do diagrama M-N- θ :

²⁷ Todos os demais tipos de diagrama M-N- θ , inclusive o de seção simétrica, são casos particulares deste.

- a) na sua forma mais geral, o diagrama M-N- θ é descontínuo, conforme mostra a curva 1 da figura 5.9; a razão desta descontinuidade reside no fato de, para uma faixa de valores de curvatura em torno de $\theta=0$ ($[-\theta', +\theta']$), não haver α que possibilite igualar β_r ao β_s dado, conforme explicado no item 5.5; novamente, qualquer algoritmo que não prevê que o diagrama M-N- θ pode assumir as formas explicadas acima, pode não funcionar com determinados problemas, conforme comentado no item 1.4
- b) quando se tem diagrama M-N- θ descontínuo, o primeiro trecho do diagrama (o trecho localizado na região de θ negativo) pode, dependendo da direção imposta para o momento resistente (β_r), ser muito inclinado (as vezes, quase vertical);
- c) a característica de convexidade do diagrama M-N- θ se fez presente na maioria absoluta dos casos; em alguns poucos casos, encontrou-se pequenos trechos levemente côncavos, quase imperceptíveis; ressalta-se que, no caso de diagrama descontínuo, a característica de convexidade é válida para os dois trechos da curva;
- d) em diagrama M-N- θ descontínuo, os pontos 'A' e 'B' (fig. 5.9) tem a mesma ordenada (M_r) e abscissas de mesmo módulo mas com sinais trocados;
- e) em diagrama M-N- θ descontínuo (fig.5.9), a inclinação (derivada primeira) do diagrama no ponto 'A' é diferente da do 'B';
- f) observe que, para seção assimétrica, quando age somente força normal na seção ($M_{sx}=M_{sy}=0$), o estado de deformação exigido para o equilíbrio não é o uniforme; o processo iterativo deve ser capaz de resolver problemas com esta característica.

Analisado o comportamento da curva $|M_r|=f(\theta)$, resta escolher o mais adequado método numérico iterativo de cálculo de zeros de funções ($y=|M_r|(\theta)-|M_s|$). Conforme explicado na observação 'c' acima, o diagrama M-N- θ é convexo; isto habilita o uso do Método da Secante, pois este método tem convergência garantida para este tipo de função e, além disso, é um método de convergência muito rápida. Porém, quando se tem diagrama descontínuo, o primeiro trecho do diagrama pode ser muito inclinado (observação 'b' acima), o que dificulta a implementação do Método da Secante neste trecho do diagrama (é difícil conseguir duas aproximações iniciais que garantam a convergência). propõe-se, então:

- diagramas contínuos e porção positiva de diagramas descontínuos → Método da Secante;
- porção negativa ($\theta < 0$) de diagramas descontínuos → Método dos Intervalos Encaixantes.

Para se conseguir as duas aproximações iniciais requeridas pelo Método da Secante, pode ser utilizado o Método dos Intervalos Encaixantes.

Quanto à verificação se ocorre ou não ruptura da seção (estado de deformação ultrapassa o ELU), observe a figura 5.10: se, durante o processo iterativo (Método da Secante), obtém-se um módulo do momento resistente menor que o solicitante ($|M_r| < |M_s|$) muito embora o estado de deformação já tenha ultrapassado o ELU (eq. 3.26 e 3.27), a capacidade da seção é dada como esgotada e o processo iterativo é finalizado.

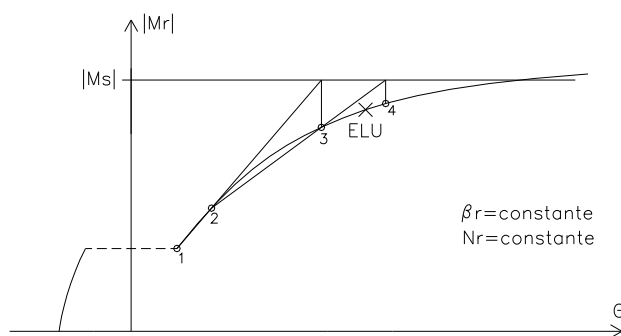


FIGURA 5.10 - Processo iterativo x ELU.

Esquemáticamente, então, o algoritmo do processo iterativo para encontrar θ que iguale $|M_r|$ a $|M_s|$ fica:

- 1) definições iniciais: precisão $x=0,001$ ($0,001 \cdot 1000h/r$); precisão $y=0,01\%$; estes valores de precisão se mostraram ideais no decorrer de testes numéricos
- 2) ROMPEU='não' (ROMPEU assume 'sim' ou 'não' caso o estado de deformação necessário ao equilíbrio ultrapasse ou não o ELU)
- 3) cálc. do ponto 'B' (ver fig. 5.9): Método dos Intervalos Encaixantes
 - 3.1) $x_b=0$; passo=0,2
 - 3.2) $\theta=x_b \rightarrow$ cálc. $|M_{rb}| = \sqrt{M_{rx}^2 + M_{ry}^2}$: processo iterativo para encontrar α que iguale β_r a β_s , item 5.5
 - 3.3) se EXISTE='não' $\rightarrow x_b=x_b+\text{passo}$, volta-se ao passo 3.2
 - 3.4) refinamento do processo: se passo=0,2 \rightarrow passo=passo/10, $x_b=x_b-0,2$, volta-se ao passo 3.2
- 4) se $|M_s| < |M_{rb}|$: a resposta está no trecho negativo do diagrama M-N- θ (ver fig. 5.9) \rightarrow Método dos Intervalos Encaixantes
 - 4.1) $x_0=-x_b$; passo=0,1
 - 4.2) $\theta=x_0 \rightarrow$ cálc. $|M_r| = \sqrt{M_{rx}^2 + M_{ry}^2}$: processo iterativo para encontrar α que iguale β_r a β_s , item 5.5 (logicamente, calcular também ε_{CG} e α)

- 4.4) se $|M_s| < |M_r| \rightarrow x_0 = x_0 - \text{passo}$, volta-se ao passo 4.2
- 4.4) refinamento do processo: se $\text{passo} = 0,1 \rightarrow \text{passo} = \text{passo}/10$, $x_0 = x_0 + 0,2$, volta-se ao passo 4.2
- 4.5) fim do processo iterativo
- 5) se $|M_s| > |M_{rb}|$: a resposta está no trecho positivo do diagrama M-N- θ (ver fig. 5.9) \rightarrow Método da Secante
- 5.1) $x_0 = x_b$; $y_0 = |M_{rb}| - |M_s|$
- 5.2) $x_1 = x_0 + 1$
- 4.3) $\theta = x_1 \rightarrow$ cálc. $y_1 = |M_r| - |M_s|$, com $|M_r| = \sqrt{M_{rx}^2 + M_{ry}^2}$: processo iterativo para encontrar α que iguale β_r a β_s , item 5.5 (logicamente, calcular também ε_{CG} e α)
- 5.4) $x_2 = x_1 - y_1 \frac{x_1 - x_0}{y_1 - y_0}$
- 5.5) $\theta = x_2 \rightarrow$ cálc. $y_2 = |M_r| - |M_s|$, com $|M_r| = \sqrt{M_{rx}^2 + M_{ry}^2}$: processo iterativo para encontrar α que iguale β_r a β_s , item 5.5 (logicamente, calcular também ε_{CG} e α)
- 5.6) se o estado de deformação ultrapassa o ELU (eq. 3.26 e 3.27) \rightarrow ROMPEU='sim'; fim do processo iterativo
- 5.7) se $(x_2 - x_1) \leq \text{precisão } x \rightarrow$ fim do processo iterativo
- 5.8) se $y_2 > |M_s| \cdot \text{precisão } y \rightarrow x_0 = x_1$; $y_0 = y_1$; $x_1 = x_2$; $y_1 = y_2$; volta-se ao passo 5.4
- 5.9) fim do processo iterativo: tem-se o estado de deformação da seção.

A título de ilustração, a seguir estão alguns diagramas M-N- θ obtidos com dados reais. Em todos os diagramas, a seção transversal é a da figura 5.2 e a força normal resistente é $N_r = 4046 \text{ kN}$ ($\nu = 0,7$). O que varia de diagrama para diagrama é a direção imposta para o momento resistente (β_r). Em alguns diagramas, a curvatura máxima (θ) é a correspondente ao ELU de ruptura; porém, em outros diagramas, não se arbitrou θ até a ruptura por uma questão de escala do gráfico.

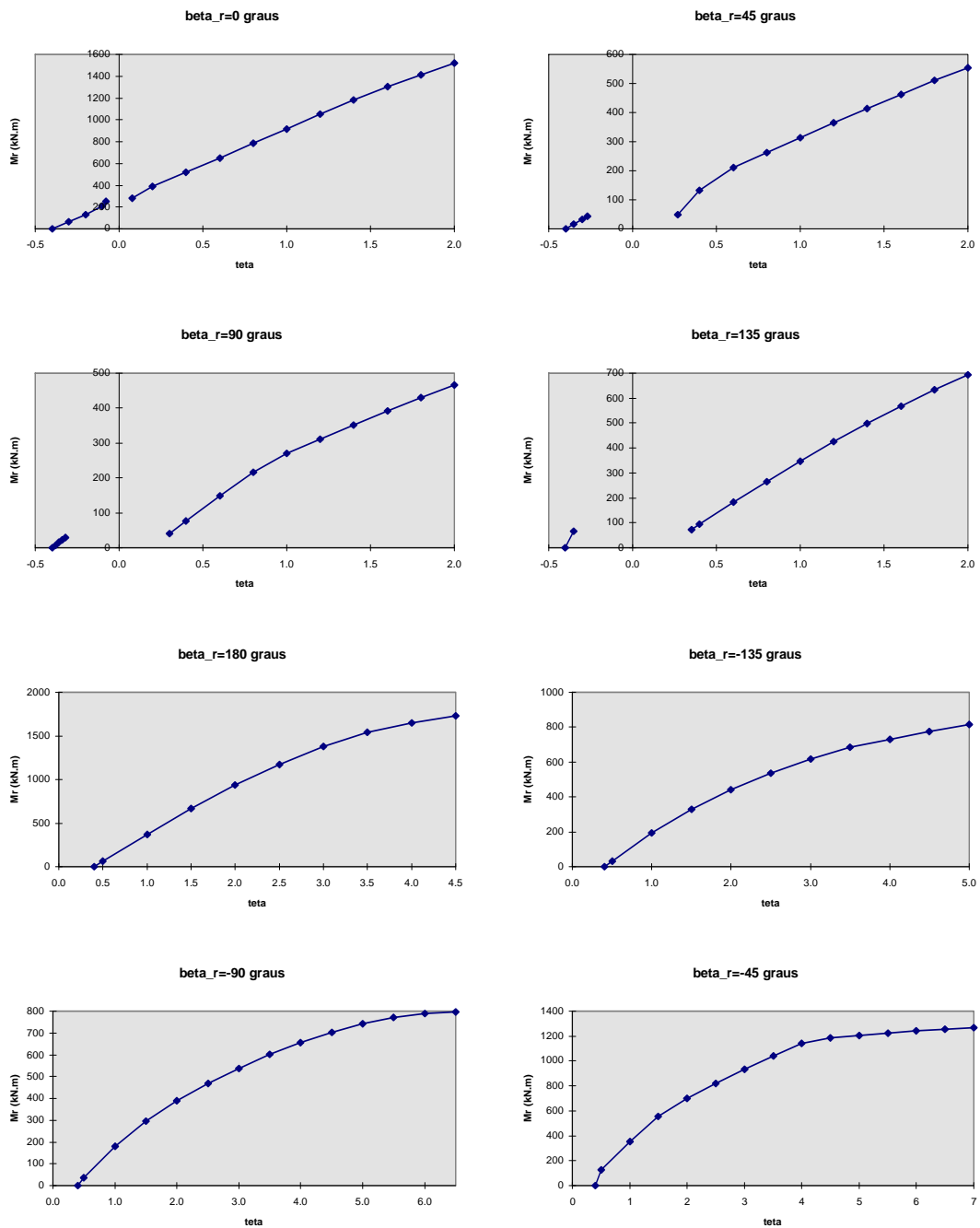


FIGURA 5.11 - Exemplos de diagrama M-N- θ .

CAPÍTULO 6 - ANÁLISE DE SEÇÕES NO ELU

6.1. INTRODUÇÃO

O presente capítulo trata da verificação e do dimensionamento de seções transversais de concreto armado no ELU de ruptura. Como se trata de análise de seção isolada, são dados do problema os esforços solicitantes totais que agem na seção (soma dos esforços de primeira ordem, de segunda ordem, acidentais, equivalentes à fluência, etc.).

Quanto ao efeito da fluência, salienta-se que a teoria linear de fluência também é adotada aqui (cálculo dos esforços resistentes conforme o capítulo 4). Porém, como se trata de análise de seção isolada, tal teoria implica somente em uma alteração na distribuição do esforço resistente total entre as parcelas resistentes devidas ao concreto e ao aço, e não em um aumento dos esforços solicitantes. Portanto, o uso da teoria linear de fluência não desabilita o estudo e quantificação do efeito da fluência no pilar como um todo (aumento dos esforços solicitantes na seção em estudo). Para tal, pode ser utilizado o conceito de excentricidade equivalente de fluência (veja, por exemplo, FUSCO(1981)).

6.2. VERIFICAÇÃO DA CAPACIDADE RESISTENTE DE SEÇÕES

6.2.1. APRESENTAÇÃO DO PROBLEMA

O objeto de estudo deste item é: dados os esforços solicitantes de uma seção definida geométrica e mecanicamente (portanto, a área das barras da armadura devem ser conhecidas), calcular se a seção sofre ou não ruptura. Esquemáticamente:

$$\left\{ \begin{array}{l} \text{seção} \\ N_s \\ M_{sx} \\ M_{sy} \end{array} \right\} \Rightarrow \text{seção rompe?}$$

Para se analisar se uma seção sofre ou não ruptura, basta calcular o máximo momento fletor a que a seção pode resistir²⁸, para, após isso, comparar com o momento solicitante dado. Caso o momento solicitante dado seja interno ao diagrama de interação da seção (fig. 6.1), a seção é considerada como não rompida. Na grande maioria dos casos, o diagrama de interação da seção assume a configuração exemplificada na figura 6.1a; neste caso, basta que $|M_s| \leq |M_r|$ para que a seção seja considerada como não rompida. Porém, em alguns casos (seções assimétricas com força normal solicitante próxima à força normal resistente última da seção), o diagrama de interação da seção pode assumir a configuração exemplificada na figura 6.1b; neste caso, é necessário o cálculo dos dois módulos de momento resistente que atendem às condições $N_r=N_s$ e $\beta_r=\beta_s$, aqui chamados de $|M_{r1}|$ e $|M_{r2}|$ (fig. 6.1b); para que a seção seja considerada como não rompida, é necessário que $|M_{r1}| \leq |M_s| \leq |M_{r2}|$. Observe que o diagrama de interação da figura 6.1a é um caso particular do diagrama da figura 6.1b, onde $|M_{r1}|=0$ e $|M_{r2}|=|M_r|$.

²⁸ Obedecendo-se às condições $N_R=N_S$ e $\beta_R=\beta_S$.

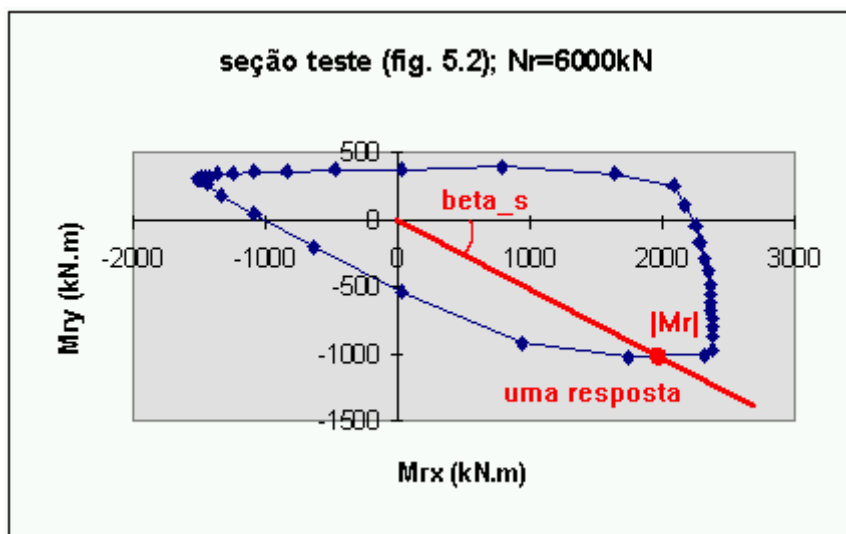


FIGURA 6.1a

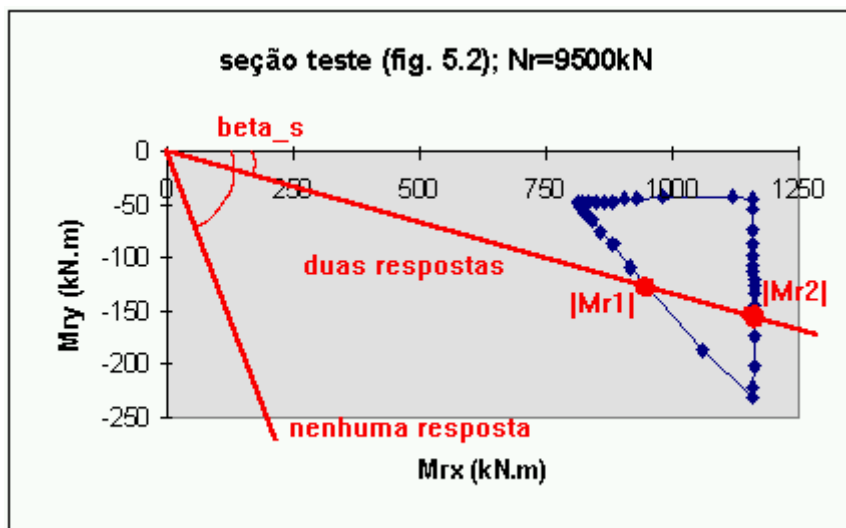


FIGURA 6.1b

FIGURA 6.1 - Andamentos típicos do diagrama de interação no ELU.

Isto posto, para se analisar se uma seção sofre ou não ruptura, basta que se calcule $|M_{r1}|$ e $|M_{r2}|$; esquematicamente:

$$\left\{ \begin{array}{l} \text{seção} \\ N_s \\ M_{sx} \\ M_{xy} \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} |M_{r1}| \\ |M_{r2}| \end{array} \right\}, \text{ tal que } \left\{ \begin{array}{l} N_r = N_s \\ \beta_{r1} = \beta_{r2} = \beta_s \end{array} \right.$$

Analogamente ao problema do capítulo anterior, o cálculo de $|M_{r1}|$ e $|M_{r2}|$ não é um problema de solução direta, pois não se sabe, *a priori*, qual a direção da linha neutra que possibilita $\beta_r = \beta_s$. Deve-se lançar mão, então, de processos iterativos.

Em última análise, o problema abordado neste item consiste em um caso particular do problema abordado no capítulo 5 (o estado de deformação deve obedecer ao ELU de ruptura - domínios de deformação). Como tal, este item consiste em adaptar os algoritmos daquele capítulo ao problema em questão. Portanto, para o entendimento deste item, é necessário o estudo prévio do capítulo 5. Também é necessário o estudo prévio do Anexo A - Métodos numéricos para o cálculo de zeros de funções.

6.2.2. PROCESSO ITERATIVO PARA VERIFICAÇÃO DE SEÇÕES

No item 5.3, viu-se que o processo iterativo indireto aplicado àquele problema é composto por três ramos independentes (processos iterativos independentes). Porém, a consideração de a seção estar no ELU, por si só elimina um dos ramos, ficando então:

- 1) arbitra-se α de 0 a 360°
- 2) arbitra-se ε_{CG}
- 3) calcula-se θ que faz com que o estado de deformação obedeça ao ELU \rightarrow eq. 3.2.2
- 4) calcula-se N_r , M_{rx} e M_{ry} (método exposto no capítulo 4)
- 5) se $N_r \neq N_s \rightarrow$ volta-se ao passo 2
- 6) se $\beta_r = \beta_s \rightarrow$ calcula-se $|M_{r1}|$ ou $|M_{r2}|$, conforme o caso
- 7) volta-se ao passo 1 até que se tenha arbitrado α de 0 a 360°

ou, dando nome a cada um dos processos iterativos independentes, tem-se (fig. 6.2):

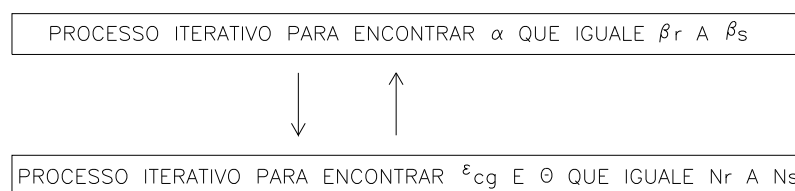


FIGURA 6.2 - Esquema geral do processo iterativo para a verificação de seções.

Nos itens seguintes, cada um dos dois processos iterativos são discutidos minuciosamente.

6.2.3. PROCESSO ITERATIVO PARA ENCONTRAR ε_{CG} E θ QUE IGUALE N_r A N_s

O problema a ser resolvido neste item é: dada a seção totalmente definida (sistema Oxy), a força normal solicitante (N_s) e a direção da linha neutra (α), calcular a deformação da origem do sistema de coordenadas (ε_{CG}) e a curvatura majorada adimensional (θ) que fazem com que a força normal resistente (N_r) se iguale à força normal solicitante dada (N_s)²⁹. Esquemáticamente, tem-se:

$$\left\{ \begin{array}{l} \text{seção} \\ \alpha \\ N_s \end{array} \right\} \Rightarrow \{ \varepsilon_{CG} \text{ e } \theta, \text{ tal que: } N_r = N_s \text{ e estado de deformação no ELU } \}$$

A resposta do problema exige um cálculo por tentativas:

- 1) arbitra-se ε_{CG}
- 2) calcula-se θ que faz com que o estado de deformação obedeça ao ELU \rightarrow eq. 3.2.2
- 3) calcula-se N_r , M_{rx} e M_{ry} (método exposto no capítulo 4)
- 4) se $N_r \neq N_s \rightarrow$ volta-se ao passo 1

Para otimizar este processo, deve-se lançar mão de um método numérico iterativo de cálculo de zeros de funções. A função, no caso, é $y=N_r(\varepsilon_{CG},\theta)-N_s$. O comportamento típico de tal função é o mesmo da função estudada no item 5.4. Portanto, o método numérico escolhido é o mesmo daquele item (Método da Posição Falsa³⁰).

A seguir, está o algoritmo proposto para o processo iterativo para encontrar ε_{CG} e θ que igualem N_r a N_s :

- 1) definições iniciais: precisão $x=0,000001\%$ e precisão $y=0,01\%$ (estes valores de precisão se mostraram ideais no decorrer de testes numéricos)
- 2) rotação do sistema de coordenadas Oxy para o Ouv (LN paralela ao eixo u): eq.4.1
- 3) determinação do intervalo inicial [a,b]: ε_{CG} estimados inicialmente com base nos estados de compressão e de tração uniformes
 - 3.1) $\theta=0 \rightarrow$ cálc. de $\varepsilon_{CG,MÍNIMO}$ e $\varepsilon_{CG,MÁXIMO}$: eq. 3.26 e 3.27
 - 3.2) cálc. de $N_{r,min}$ e $N_{r,max}$: método exposto no capítulo 4

²⁹ A materialização computacional deste item é a sub-rotina “PROCURA_e_CG_TETA_ELU”, constante do Apêndice 1.

³⁰ Ver Anexo A ou RUGGIERO & LOPES(1988).

- 3.3) intervalo inicial: $a=\varepsilon_{CG,MÍNIMO}$; $ya=N_{r,min}-N_s$; $b=\varepsilon_{CG,MÁXIMO}$; $yb=N_{r,max}-N_s$
- 4) Cálculo da raiz: Método da Posição Falsa
- 4.1) $M=ya$; $yx=ya$
- 4.2) $x=(a.yb-b.ya)/(yb-ya)$
- 4.3) $\varepsilon_{CG}=x \rightarrow$ cálc. θ : eq. 3.22
- 4.4) cálc. N_r , M_{ru} , M_{rv} : método exposto no capítulo 4; $yx=N_r-N_s$
- 4.5) se $M.yx \geq 0$: $a=x$; $ya=yx$
- 4.6) se $M.yx < 0$: $b=x$; $yb=yx$
- 4.7) se $|b-a| \leq$ precisão x :
- 4.7.1) $\varepsilon_{CG}=(a+b)/2 \rightarrow$ cálc. θ : eq. 3.22
- 4.7.2) cálc. N_r , M_{ru} , M_{rv} : método exposto no capítulo 4
- 4.7.3) cálc. M_{rx} e M_{ry} : eq. 4.2
- 4.7.4) fim do processo
- 4.8) se $yx >$ precisão $y.N_s$: volta-se ao passo 4.2
- 4.9) cálc. M_{rx} e M_{ry} : eq. 4.2
- 4.10) fim do processo

6.2.4. PROCESSO ITERATIVO PARA ENCONTRAR α QUE IGUALE β_r A β_s

O problema a ser resolvido neste item é: dada a seção totalmente definida, a força normal solicitante (N_s) e a direção do momento fletor solicitante (β_s), calcular as direções da linha neutra (α_1 e α_2) que fazem com que as direções dos momentos resistentes (β_{r1} e β_{r2}) sejam iguais a direção do momento solicitante dado (β_s)³¹, veja a fig. 6.1. Logicamente, a condição $N_r=N_s$ também deve ser atendida, mas esta é encargo do item 6.3.2. Esquemáticamente, tem-se:

$$\left\{ \begin{array}{l} \text{seção} \\ \beta_s \end{array} \right\} \Rightarrow \left\{ \alpha_1 \text{ e } \alpha_2, \text{ tal que } \beta_{r1} = \beta_{r2} = \beta_s \right\}$$

A resposta do problema exige um cálculo por tentativas:

- 1) arbitra-se α de 0 a 360°
- 2) cálc. $\beta_r \rightarrow$ item 6.2.3
- 3) se $\beta_r=\beta_s \rightarrow$ calcula-se $|M_{r1}|$ ou $|M_{r2}|$, conforme o caso

³¹ A materialização computacional deste item é a sub-rotina “VERIFIC_ELU”, constante do Apêndice 1.

4) volta-se ao passo 1 até que se tenha arbitrado α de 0 a 360°

Para otimizar este processo, deve-se lançar mão de um método numérico iterativo de cálculo de zeros de funções. A função, no caso, é $y = \beta_r(\alpha) - \beta_s$. O andamento dessa curva já foi estudado no item 5.5. Portanto, os métodos utilizados são os mesmos já descritos naquele item.

A seguir, está o algoritmo proposto para o processo iterativo para encontrar α que iguale β_r a β_s :

- 1) definições iniciais: precisão $x = \text{precisão } y = 0,0001 \text{ rad } (0,0057 \text{ graus})$; estes valores de precisão se mostraram ideais no decorrer de testes numéricos
- 2) EXISTE='sim': EXISTE assume 'sim' ou 'não' caso exista ou não um α que possibilite igualar β_r a β_s
 ETAPA=1: ETAPA indica o ponto que se está calculando (problemas com 2 respostas, veja fig.6.1)
 N_RESPOSTA=2: N_RESPOSTA indica o número de respostas que o problema apresenta (veja fig. 6.1); inicialmente, assume-se que o problema tem 2 respostas
- 3) se a seção for duplamente simétrica:
 ETAPA=2
 N_RESPOSTA=1
- 4) cálc. do 'a' inicial: como tentativa inicial, coloca-se a LN na direção perpendicular à do momento solicitante
 - 4.1) $a = \beta_s - \pi/2$
 - 4.2) cálc. y_a :
 - 4.2.1) $\alpha = a \rightarrow$ cálc. $y_a = \beta_r - \beta_s$, com $\beta_r = \arctan(M_{ry}/M_{rx})$: processo iterativo para encontrar ε_{CG} e θ que iguale N_r a N_s (item 6.2.3)
 - 4.2.2) se $y_a < -\pi \rightarrow y_a = y_a + 2\pi$
 - 4.2.3) se $y_a > +\pi \rightarrow y_a = y_a - 2\pi$
 - 4.3) se $|y_a| \leq \text{precisão } y \rightarrow$ é o ponto 2, veja fig.6.1
 - 4.3.1) se a seção for duplamente simétrica: fim do processo
- 5) preparativos para o cálculo dos b's iniciais:
 - 5.1) determinação se α deve aumentar (variar no sentido trigonométrico) ou diminuir para poder encontrar o α_{RAIZ2} (ponto 2, ver fig. 6.1):

- 5.1.1) se $\text{sen}(\beta_r - \beta_s) < 0 \rightarrow \text{sinal} = +1$ (α deve aumentar)
- 5.1.2) se $\text{sen}(\beta_r - \beta_s) > 0 \rightarrow \text{sinal} = -1$ (α deve diminuir)
- 5.2) determinação do passo a ser utilizado:
- 5.2.1) se a seção for duplamente simétrica $\rightarrow \text{passo} = \text{sinal} \cdot \pi / 12$ (15 graus)
- 5.2.1) se a seção não for duplamente simétrica $\rightarrow \text{passo} = \text{sinal} \cdot \pi / 36$ (5 graus)
- 5.3) no início do processo, 'b' deve ser igual a 'a': $b = a$; $y_b = y_a$
- 5.4) ao longo do processo, 'a' inicial será modificado: guardar o cálculo de 'a' feito com $\alpha = \beta_s - \pi / 2$: $a_0 = a$; $y_{a0} = y_a$
- 6) cálc. dos b's iniciais: (Método dos Intervalos Encaixantes)
- 6.1) $a = b$; $y_a = y_b$
- 6.2) $b = b + \text{passo}$
- 6.3) se $|b - a_0| \geq \pi$: α já foi incrementado em 180° e não foi encontrada raiz
- 6.3.1) se ETAPA=1 \rightarrow não há raiz (problema sem resposta, veja a fig. 6.1)
 $\rightarrow \text{EXISTE} = \text{não} \rightarrow$ fim do processo
- 6.3.2) se ETAPA=2 \rightarrow só ha uma raiz (problema com uma só resposta, veja a fig. 6.1)
 \rightarrow a raiz é a 'a' inicial
- 6.4) cálc. y_b :
- 6.4.1) $\alpha = b \rightarrow$ cálc. $y_b = \beta_r - \beta_s$, com $\beta_r = \arctan(M_{ry} / M_{rx})$: processo iterativo para encontrar ε_{CG} e θ que iguale N_r a N_s (item 6.2.3)
- 6.4.2) se $y_b < -\pi \rightarrow y_b = y_b + 2\pi$
- 6.4.3) se $y_b > +\pi \rightarrow y_b = y_b - 2\pi$
- 6.5) se $y_a \cdot y_b > 0 \rightarrow$ volta-se ao passo 6.1
- 6.6) se ETAPA=1: o intervalo encontrado contém o ponto 2 (veja a fig. 6.1)
- 6.6.1) ETAPA=2, vá para 6.1 (encontrar o outro intervalo com a outra raiz)
- 6.7) se ETAPA=2: o intervalo encontrado contém o ponto 1 (veja a fig. 6.1)
- 7) cálculo da raiz (Método da Posição Falsa): para cada um dos intervalos que contém raiz (pode ser 1, 2 ou nenhum intervalo, veja a fig.6.1), executar a rotina a seguir
- 7.1) $M = y_a$; $y_x = y_a$
- 7.2) se $M \cdot y_x \geq 0$: $a = x$; $y_a = y_x$
- 7.3) se $M \cdot y_x < 0$: $b = x$; $y_b = y_x$
- 7.4) se $|b - a| \leq \text{precisão } x$:
- 7.4.1) $\alpha = (a + b) / 2 \rightarrow$ cálc. N_r , M_{ru} , M_{rv} : processo iterativo para encontrar ε_{CG} e θ que iguale N_r a N_s (item 6.2.3)
- 7.4.2) fim do processo

$$7.6) x = (a \cdot y_b - b \cdot y_a) / (y_b - y_a)$$

7.7) cálc. y_x :

7.7.1) $\alpha = x \rightarrow$ cálc. $y_x = \beta_r - \beta_s$, com $\beta_r = \arctan(M_{ry}/M_{rx})$: processo iterativo para encontrar ε_{CG} e θ que iguale N_r a N_s (item 6.2.3)

7.7.2) se $y_x < -\pi \rightarrow y_x = y_x + 2\pi$

7.7.3) se $y_x > +\pi \rightarrow y_x = y_x - 2\pi$

7.8) se $y_x >$ precisão $y \rightarrow$ volta-se ao passo 7.2

7.9) fim do processo

6.3. DIMENSIONAMENTO DA ARMADURA DE SEÇÕES

O problema a ser resolvido neste item é: dada a seção totalmente definida (a menos da área das barras de aço) e os esforços solicitantes, calcular a menor taxa mecânica de armadura (ω) que propicia a que a seção resista aos esforços solicitantes dados sem romper³².

Supõe-se que todas as barras da armadura tem a mesma área. Esquemáticamente, tem-se:

$$\left\{ \begin{array}{l} \text{secao (a menos de } \omega) \\ N_s, M_{sx}, M_{sy} \end{array} \right\} \Rightarrow \{ \omega_{\min}, \text{ tal que a secao nao rompa} \}$$

A resposta do problema exige um cálculo por tentativas:

- 1) arbitra-se ω (taxa mecânica de armadura, $\omega = \frac{A_{s,\text{total}} \cdot f_y}{A_c \cdot f_c}$)
- 2) verifica-se se a seção rompe ou não \rightarrow item 6.2
- 3) se a seção rompe \rightarrow volta-se ao passo 1

Para otimizar este processo, pode-se lançar mão de um método numérico iterativo de cálculo de zeros de funções³³. No caso, a Método da Bissecção se adapta perfeitamente. Conforme explica-se no Anexo A, o Método da Bissecção necessita, para começar o processo iterativo, que se determine um intervalo inicial $[a,b]$ que contenha internamente a raiz procurada ($a \leq \omega \leq b$). Para se encontrar esse intervalo, pode-se lançar mão do Método dos Intervalos Encaixantes.

³² A materialização computacional deste item é a sub-rotina “DIM_ELU”, constante do Apêndice 1.

³³ Ver Anexo A ou RUGGIERO & LOPES(1988).

Apresenta-se a seguir o algoritmo proposto para o dimensionamento da armadura de seções:

- 1) $a=0$ ('a' e 'b' representam taxa mecânica de armadura)
- 2) $\omega=a \rightarrow$ cálc. se a seção rompe ou não (item 6.2)
- 3) se a seção rompe, $y_a=-1$; senão, $y_a=+1$
- 4) se a seção não rompeu: fim do processo
- 5) $b=1$
- 6) $\omega=b \rightarrow$ cálc. se a seção rompe ou não (item 6.2)
- 7) se a seção rompe, $y_b=-1$; senão, $y_b=+1$
- 8) se $y_a \cdot y_b > 0$: o intervalo $[a,b]$ encontrado ainda não contém raiz \rightarrow Método dos Intervalos Encaixantes
 - 8.1) $a=b$; $y_a=y_b$
 - 8.2) $b=b+1$
 - 8.3) $\omega=b \rightarrow$ cálc. se a seção rompe ou não (item 6.2)
 - 8.4) se a seção rompe, $y_b=-1$; senão, $y_b=+1$
 - 8.5) volta-se ao passo 8
- 9) cálc. da raiz: Método da Bissecção
 - 9.1) $M=y_a$
 - 9.2) $x=(a+b)/2$
 - 9.3) $\omega=x \rightarrow$ cálc. se a seção rompe ou não (item 6.2)
 - 9.4) se a seção rompe, $y_x=-1$; senão, $y_x=+1$
 - 9.5) se $M \cdot y_x > 0$: $a=x$; $y_a=y_x$;
senão: $b=x$; $y_b=y_x$
 - 9.6) se $(b-a) > \text{TOLERÂNCIA}$: volta-se ao passo 9.2.

CAPÍTULO 7 - ANÁLISE DA ESTABILIDADE DE PILARES

7.1. INTRODUÇÃO

Neste capítulo, apresenta-se a análise da estabilidade de pilares isolados de concreto armado solicitados por flexão composta oblíqua. Trata-se, portanto, de análise não-linear física (o material não segue a lei de Hooke) e geométrica (análise em 2ª ordem). Para tal, utiliza-se o método do equilíbrio e o método geral com o processo exato e com o processo do pilar padrão.

7.2. VERIFICAÇÃO DA ESTABILIDADE PELO MÉTODO DO EQUILÍBRIO COM O PROCESSO EXATO

7.2.1. PROCESSO EXATO

Neste item, aplica-se o método do equilíbrio com o processo exato na verificação da estabilidade de pilares. Trata-se de um processo bastante geral, que pode ser aplicado à pilares de grande esbeltez, pilares de seção transversal variável ao longo do comprimento, pilares com distribuição qualquer de esforços solicitantes ao longo do comprimento, etc.

7.2.2. DISCRETIZAÇÃO DO PILAR

O processo exato necessita que se divida o pilar em vários segmentos, através da discretização do pilar em seções (fig. 7.1). O processo exato calcula as curvaturas, os deslocamentos transversais e os momentos de segunda ordem somente nestas seções. A verificação da ruptura também é feita somente nestes pontos. A hipótese simplificadora do processo exato é a de que a curvatura varia linearmente entre duas seções consecutivas. Portanto, a precisão dos resultados obtidos será tanto maior quanto maior for a quantidade de seções utilizadas na discretização do pilar.

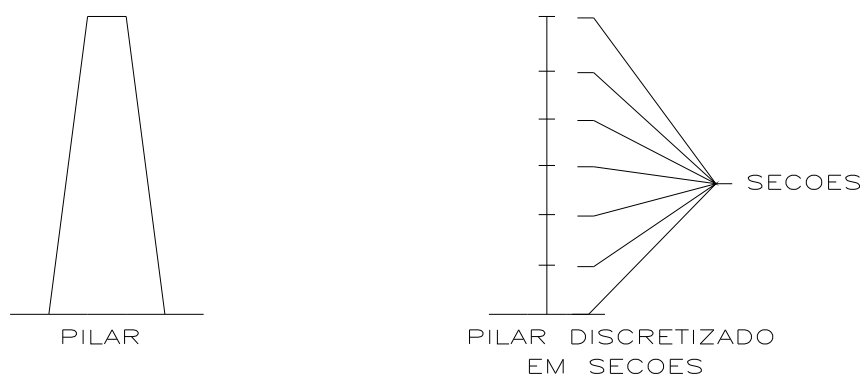


FIGURA 7.1 - Discretização do pilar em seções transversais.

Quando se tem pilares com variação brusca de seção transversal ou de esforços solicitantes de primeira ordem, deve-se discretizar o pilar com uma seção antes e outra após a variação brusca (fig. 7.2).

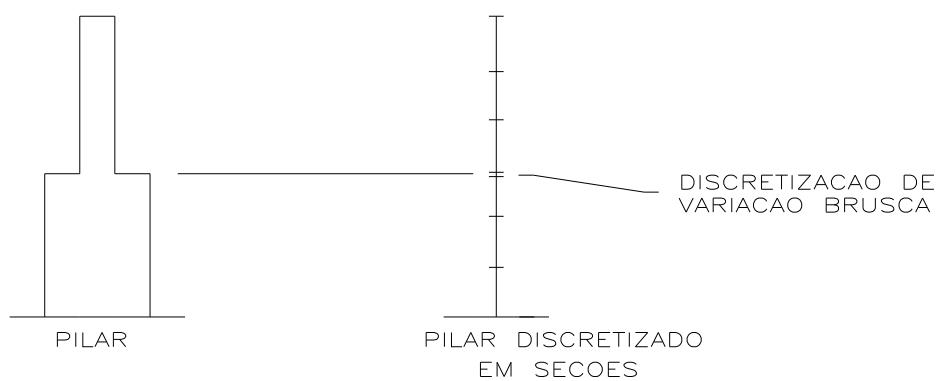


FIGURA 7.2 - Discretização de pilar com variação brusca de seção ou de esforços.

Quanto aos tipos de vinculações, neste trabalho admite-se pilares (veja a fig. 7.3):

- a) biapoiados
- b) rigidamente engastados na base
- c) elasticamente engastados na base



FIGURA 7.3 - Vinculação admitidas para os pilares.

Observe que um pilar pode ter um tipo de vinculação na direção x e outro tipo na direção y.

A automatização do processo exato exige que se convençione a maneira como se discretiza o pilar. Neste trabalho, tal convenção é a seguinte (ver fig. 7.4):

- o eixo do pilar deve coincidir com o eixo z;
- o pilar deve ser discretizado com seções numeradas de 1 a $N_{\text{seção}}$; a cada seção corresponde a sua respectiva cota z;
- a seção 1 deve ter $z = 0$; as demais seções devem ter $z > 0$; duas seções podem ter o mesmo z (caso de variação brusca de seção ou de esforços solicitantes);
- caso o pilar seja engastado, a seção de engaste deve ser a de número 1;
- cada seção transversal deve ser discretizada conforme indicado no item 2.7.

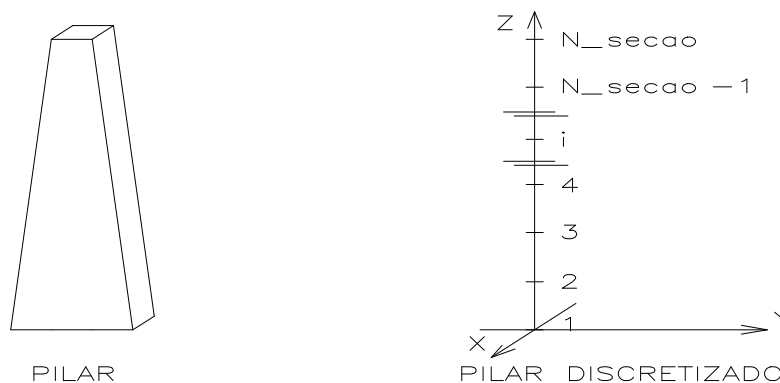


FIGURA 7.4 - Convenção para discretização do pilar.

7.2.3. CÁLCULO DOS DESLOCAMENTOS TRANSVERSAIS

Um passo básico do processo exato é o cálculo dos deslocamentos transversais do pilar, partindo-se das curvaturas calculadas nas várias seções. O cálculo dos deslocamentos

transversais do eixo de um pilar fletido é feito através de uma dupla integração numérica da equação diferencial da curvatura (eq. 2.1) ao longo do comprimento do pilar.

Como se trata de flexão composta oblíqua, existem deslocamentos transversais tanto na direção x quanto na direção y . Estes deslocamentos são calculados separadamente. FUSCO (1981) demonstra que, tendo-se a curvatura na direção perpendicular à da linha neutra ($1/r_\alpha$) e a direção da linha neutra (α), pode-se calcular a curvatura nas direções x e y através de:

$$\frac{1}{r_x} = \frac{1}{r_\alpha} \cdot \text{SEN}(\alpha)$$

$$\frac{1}{r_y} = \frac{1}{r_\alpha} \cdot \text{COS}(\alpha)$$
(7.1)

Seja, então, um pilar discretizado conforme o item 7.2.2. Dadas as curvaturas nas várias seções (fig. 7.5), tem-se que as rotações (φ) nas várias seções são dadas pela integral das curvaturas (eq. 2.1).

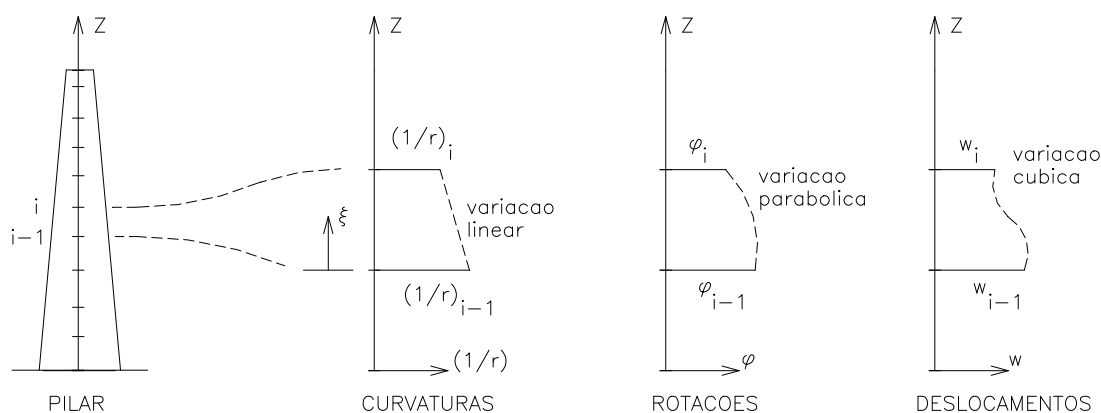


FIGURA 7.5 - Curvaturas, rotações e deslocamentos do eixo do pilar.

Fazendo-se uma mudança de coordenadas (fig. 7.5) para facilitar a integração, vem:

$$\xi = z - z_{i-1}$$
(7.2)

$$\xi_i = z_i - z_{i-1},$$
(7.3)

onde: ξ_i = coordenada ξ na seção i ;

z_i = coordenada z na seção i .

Da hipótese de que a curvatura varia linearmente entre duas seções consecutivas, vem:

$$\frac{1}{r} = a \cdot \xi + b, \quad (7.4)$$

$$\text{com: } \begin{cases} a = \frac{\frac{1}{r_i} - \frac{1}{r_{i-1}}}{z_i - z_{i-1}} \\ b = \frac{1}{r_{i-1}} \end{cases} \quad (7.5)$$

Rotações entre z_{i-1} e $z_i \rightarrow$ integral das curvaturas (eq. 7.4):

$$\varphi = \int \frac{1}{r} d\xi$$

da eq. 7.4, vem:

$$\varphi = \int (a \cdot \xi + b) d\xi$$

resolvendo a integral, vem:

$$\varphi = \frac{a \cdot \xi^2}{2} + b \cdot \xi + c \quad (7.6)$$

mas,

$$\varphi(0) = \varphi_{i-1} \therefore c = \varphi_{i-1} \quad (7.7)$$

substituindo-se as eq. 7.3, 7.5 e 7.7 em 7.6, vem:

$$\varphi_i = \varphi_{i-1} + \frac{\frac{1}{r_i} + \frac{1}{r_{i-1}}}{2} (z_i - z_{i-1}), \quad (7.8)$$

onde: φ_i = rotação da seção i;

$$\frac{1}{r_i} = \text{curvatura da seção i.}$$

ou, escrevendo de outra maneira,

$$\varphi_i = \varphi_1 + \sum_{k=2}^i \frac{\frac{1}{r_k} + \frac{1}{r_{k-1}}}{2} (z_k - z_{k-1}),$$

onde: φ_1 = rotação da seção 1;

ou, escrevendo de outra maneira,

$$\varphi_i = \varphi_1 + \varphi_i' \quad (7.9)$$

$$\text{com: } \varphi_i' = \sum_{k=2}^i \frac{\frac{1}{r_k} + \frac{1}{r_{k-1}}}{2} (z_k - z_{k-1}) \quad (7.10)$$

observe que φ_i' também pode ser calculado com:

$$\begin{aligned}\varphi_1' &= 0 \\ \varphi_i' &= \varphi_{i-1}' + \frac{\frac{1}{r_i} + \frac{1}{r_{i-1}}}{2} (z_i - z_{i-1})\end{aligned}\quad (7.11)$$

Deslocamentos transversais (w) z_{i-1} e $z_i \rightarrow$ integral das rotações (eq. 7.6):

$$w = \int \varphi \cdot d\xi$$

da eq. 7.6, vem:

$$w = \int \left(\frac{a \cdot \xi^2}{2} + b \cdot \xi + c \right) d\xi$$

resolvendo a integral, vem:

$$w = \frac{a \cdot \xi^3}{6} + \frac{b \cdot \xi^2}{2} + c \cdot \xi + d \quad (7.12)$$

mas,

$$w(0) = w_{i-1} \therefore d = w_{i-1} \quad (7.13)$$

substituindo-se as eq. 7.3, 7.5, 7.7 e 7.13 em 7.12, vem:

$$w_i = w_{i-1} + \frac{\frac{1}{r_i} + 2 \frac{1}{r_{i-1}}}{6} (z_i - z_{i-1})^2 + \varphi_{i-1}' (z_i - z_{i-1}), \quad (7.14)$$

onde: w_i = deslocamento transversal da seção i ;

substituindo-se a eq. 7.9 em 7.14, vem:

$$w_i = w_{i-1} + \frac{\frac{1}{r_i} + 2 \frac{1}{r_{i-1}}}{6} (z_i - z_{i-1})^2 + \varphi_{i-1}' (z_i - z_{i-1}) + \varphi_1 (z_i - z_{i-1})$$

ou, escrevendo de outra maneira,

$$w_i = w_1 + \sum_{k=2}^i \left(\frac{\frac{1}{r_k} + 2 \frac{1}{r_{k-1}}}{6} (z_k - z_{k-1})^2 + \varphi_{k-1}' (z_k - z_{k-1}) \right) + \sum_{k=2}^i \varphi_1 (z_k - z_{k-1})$$

ou seja,

$$w_i = w_1 + \sum_{k=2}^i \left(\frac{\frac{1}{r_k} + 2 \frac{1}{r_{k-1}}}{6} (z_k - z_{k-1})^2 + \varphi_{k-1}' (z_k - z_{k-1}) \right) + \varphi_1 \cdot z_i$$

ou, escrevendo de outra maneira,

$$w_i = w_1 + \varphi_1 \cdot z_i + w_i' \quad (7.15)$$

$$\text{com: } w'_i = \sum_{k=2}^i \left(\frac{1}{r_k} + 2 \frac{1}{r_{k-1}} (z_k - z_{k-1})^2 + \varphi'_{k-1} (z_k - z_{k-1}) \right)$$

observe que w'_i também pode ser calculado com:

$$w'_1 = 0$$

$$w'_i = w'_{i-1} + \frac{1}{r_i} + 2 \frac{1}{r_{i-1}} (z_i - z_{i-1})^2 + \varphi'_{i-1} (z_i - z_{i-1}) \quad (7.16)$$

Assim, os deslocamentos transversais das várias seções de um pilar fletido podem ser calculados através das equações 7.11, 7.15 e 7.16. Os valores w_1 e φ_1 vem do tipo de vinculação do pilar (condições de contorno). Assim, pilares:

a) biapoiados:	$w_1=0$	$\varphi_1 = -\frac{w'_{N_secao}}{Z_{N_secao}}$
b) rigidamente engastados na base:	$w_1=0$	$\varphi_1=0$
c) elasticamente engastados na base:	$w_1=0$	$\varphi_1 = \frac{M_1}{k_{mola}}$,

onde: M_1 = momento fletor que solicita a seção 1 (seção da base), na direção em que se está analisando o pilar (x ou y);

k_{mola} = constante de mola da base elástica (momento necessário para causar um giro unitário), na direção em que se está analisando o pilar (x ou y).

Apresenta-se a seguir o algoritmo proposto para o cálculo dos deslocamentos transversais do eixo do pilar³⁴:

- 1) calcula-se as curvaturas nas direções x e y em todas as seções do pilar → eq. 7.1
- 2) calcula-se φ' nas direções x e y em todas as seções do pilar → eq. 7.11
- 3) calcula-se w' nas direções x e y em todas as seções do pilar → eq. 7.16
- 4) calcula-se φ_1 e w_1 nas direções x e y em todas as seções do pilar → condições de contorno do pilar
- 5) calcula-se os deslocamentos transversais nas direções x e y em todas as seções do pilar → eq. 7.15

³⁴ A materialização computacional deste item é a sub-rotina "CALC_DESLOCAMENTOS", constante do Apêndice 1.

Conforme visto, o cálculo dos deslocamentos transversais do eixo de um pilar é feito realizando-se uma dupla integração do diagrama de curvaturas do pilar. Para tal, a única hipótese feita foi a de que a curvatura varia linearmente entre duas seções consecutivas (fig. 7.5), o que implica em que a rotação varia segundo uma parábola do segundo grau. Alguns trabalhos correlatos a este, porém, realizam a dupla integração através da utilização da regra dos trapézios duas vezes; isto implica em admitir que a curvatura e a rotação variam linearmente entre duas seções consecutivas, o que causa uma desnecessária perda de precisão.

7.2.4. CÁLCULO DOS MOMENTOS DE 2ª ORDEM

Um passo básico do processo exato, é o cálculo dos momentos fletores de 2ª ordem, partindo-se dos deslocamentos transversais calculados nas várias seções do pilar. Da mesma maneira que no item anterior, o cálculo dos momentos de 2ª ordem deve ser feito separadamente nas direções x e y do pilar.

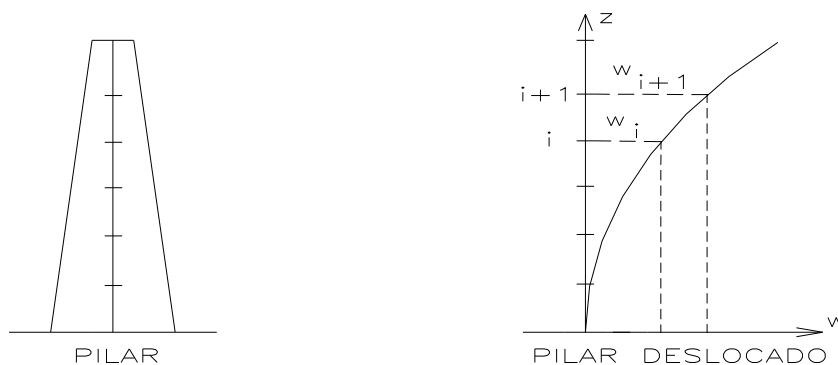


FIGURA 7.6 - Pilar deslocado.

Seja, então, um pilar discretizado conforme o item 7.2.2; dados os deslocamentos (w) e as forças normais solicitantes (N_S) nas várias seções (fig. 7.6), tem-se que os momentos de 2ª ordem ($M2$) são dados por:

$$\begin{aligned} M2_{N_secao} &= 0 \\ M2_i &= M2_{i+1} + N_{tot_{i+1}}(w_{i+1} - w_i) \end{aligned} \quad (7.17)$$

onde: $M2_{N_secao}$ = momento fletor de segunda ordem que solicita a seção $N_seção$, na direção em que se está analisando o pilar (x ou y);

$M2_i$ = momento fletor de segunda ordem que solicita a seção i , na direção em que se está analisando o pilar (x ou y);

$N_{tot_{i+1}}$ = força normal total que solicita a seção $i+1$ (veja a seguir).

Conforme explicado por FUSCO (1981), quando o pilar em análise faz parte da estrutura de contraventamento de um edifício, o cálculo do momento de segunda ordem deve ser feito como se toda a carga vertical de cada pavimento da estrutura contraventada fosse aplicada na estrutura de contraventamento, nas seções de mesmo nível de cada pavimento. Assim,

$$N_{tot,i+1} = N_{S,i+1} + N_{contrav,i+1} \quad (7.18)$$

onde: $N_{S,i+1}$ = força normal solicitante que atua na seção $i+1$ do pilar em análise;

$N_{contrav,i+1}$ = caso o pilar em análise faça parte da estrutura de contraventamento, $N_{contrav,i+1}$ é a soma das forças normais solicitantes, no nível $i+1$, dos pilares da estrutura contraventada pelo pilar em análise; caso o pilar em análise não faça parte da estrutura de contraventamento, $N_{contrav,i+1} = 0$.

7.2.5. PROCESSO ITERATIVO PARA A VERIFICAÇÃO DA ESTABILIDADE PELO MÉTODO DO EQUILÍBRIO COM O PROCESSO EXATO

O problema a ser resolvido é: dado um pilar – discretizado conforme o item 7.2.2 – totalmente definido (portanto, a área das barras da armadura devem ser conhecidas) e os esforços que solicitam o pilar, verificar se tal pilar é ou não estável e, caso seja, calcular os deslocamentos transversais do eixo do pilar³⁵. Esquemáticamente:

$$\left\{ \begin{array}{l} \text{pilar totalmente definido} \\ \left\{ \begin{array}{l} N_s \\ M_{sx} \\ M_{sy} \end{array} \right\} \text{ em todas as seções} \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} \text{pilar rompe?} \\ \text{deslocamentos transversais} \end{array} \right.$$

A verificação da estabilidade pelo método do equilíbrio é feita através de um processo iterativo. Os momentos de segunda ordem calculados em uma iteração são somados aos esforços solicitantes iniciais e utilizados para calcular os deslocamentos transversais na próxima iteração. O processo iterativo deve ser repetido até que os deslocamentos transversais nas várias seções convirjam para valores finitos; ou até que alguma seção atinja o ELU de ruptura.

³⁵ A materialização computacional deste item é a sub-rotina “VERIFIC_EXATO”, constante do Apêndice 1.

Apresenta-se a seguir o algoritmo proposto para o processo iterativo para verificação da estabilidade de pilares pelo método do equilíbrio com o processo exato:

- 1) calcula-se os momentos fletores totais que solicitam cada seção transversal (nas direções x e y): soma do momento solicitante (M_s) com o momento de 2ª ordem (M_2); na primeira iteração, $M_2=0$;
- 2) utilizando-se os momentos fletores totais, calcula-se θ (curvatura majorada adimensional), α (direção da linha neutra) e h_α (altura da seção transversal na direção perpendicular a da linha neutra) em todas as seções → item 5.6;
- 3) se qualquer uma das seções transversais atingir o ELU de ruptura, fim do processo iterativo → o pilar não é estável;
- 4) calcula-se a curvatura ($\frac{1}{r_\alpha}$) em todas as seções → eq. 2.9: $\frac{1}{r_\alpha} = \frac{\theta}{1000h_\alpha}$;
- 5) calcula-se os deslocamentos transversais de todas as seções (nas direções x e y) → item 7.2.3;
- 6) calcula-se os momentos de 2ª ordem em todas as seções (nas direções x e y) → item 7.2.4;
- 7) se, para todas as seções, os deslocamentos transversais calculados forem iguais (a menos de uma certa tolerância) aos deslocamentos transversais calculados na iteração anterior, fim do processo iterativo → o pilar é estável;
- 8) se o deslocamento transversal em alguma seção divergir (não tender a um valor finito), fim do processo iterativo → o pilar não é estável;
- 9) volta-se ao passo 1.

7.3. VERIFICAÇÃO DA ESTABILIDADE PELO MÉTODO DO EQUILÍBRIO COM O PROCESSO DO PILAR PADRÃO

7.3.1. PROCESSO DO PILAR PADRÃO

Neste item, aplica-se o método do equilíbrio com o processo do pilar padrão na verificação da estabilidade de pilares. Como se viu, o processo exato exige que, em cada iteração, calcule-se a curvatura, deslocamento transversal e momento de segunda ordem em todas as seções transversais utilizadas para discretizar o pilar, acarretando, assim, uma grande quantidade de cálculos. O processo do pilar padrão foi criado com o objetivo de

diminuir o grande número de cálculos necessários para a verificação da estabilidade de pilares. O processo do pilar padrão baseia-se na suposição de ser senoidal a elástica do pilar, o que elimina a necessidade de se discretizar o pilar com várias seções; em cada iteração, calcula-se a curvatura e o momento de segunda ordem somente em uma seção (seção de referência). Devido à hipótese de ser senoidal a elástica do pilar, a utilização do processo do pilar padrão restringe-se a pilares de seção constante (inclusive armadura) e força normal constante ao longo do comprimento do pilar.

Pilar padrão é um pilar em balanço com uma distribuição de curvaturas que provoque, na sua extremidade livre, uma flecha (w_{\max}) dada por (fig. 7.7):

$$w_{\max} = \frac{l_e^2}{10} \left(\frac{1}{r_\alpha} \right)$$

onde: l_e = comprimento de flambagem;

nas direções x e y, as flechas máximas são dadas por:

$$w_{\max,x} = \frac{l_e^2}{10} \left(\frac{1}{r_x} \right)$$

$$w_{\max,y} = \frac{l_e^2}{10} \left(\frac{1}{r_y} \right)$$

(7.19)

com $\frac{1}{r_x}$ e $\frac{1}{r_y}$ dadas pela equação 7.1

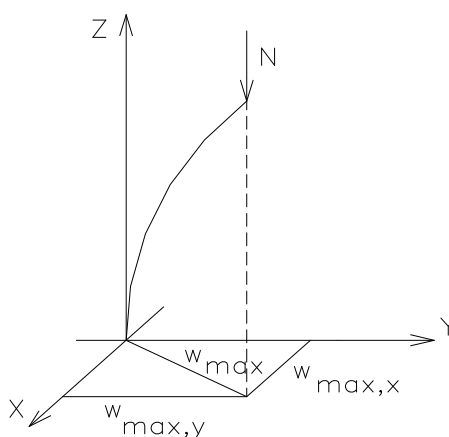


FIGURA 7.7 - Pilar padrão.

7.3.2. PROCESSO ITERATIVO PARA A VERIFICAÇÃO DA ESTABILIDADE PELO MÉTODO DO EQUILÍBRIO COM O PROCESSO DO PILAR PADRÃO

Trata-se do mesmo problema do item 7.2.5, porém adaptado ao processo aproximado do pilar padrão. Apresenta-se a seguir o algoritmo proposto para o processo iterativo para verificação da estabilidade de pilares pelo método do equilíbrio com o processo do pilar padrão³⁶:

- 1) calcula-se os momentos fletores totais que solicitam a seção transversal de referência (nas direções x e y): soma do momento solicitante (M_s) com o momento de 2ª ordem (M_2); na primeira iteração, $M_2=0$;
- 2) utilizando-se os momentos fletores totais, calcula-se θ (curvatura majorada adimensional), α (direção da linha neutra) e h_α (altura da seção transversal na direção perpendicular a da linha neutra) da seção de referência → item 5.6;
- 3) se a seção de referência atingir o ELU de ruptura: o pilar rompe → fim do processo iterativo
- 4) calcula-se a curvatura ($\frac{1}{r_\alpha}$) na seção de referência → eq. 2.9: $\frac{1}{r_\alpha} = \frac{\theta}{1000h_\alpha}$;
- 5) calcula-se os deslocamentos transversais $w_{\max,x}$ e $w_{\max,y}$ → eq. 7.19;
- 6) calcula-se os momentos de 2ª ordem na seção de referência:

$$M_{2_x} = N_s \cdot W_{\max,x}$$

$$M_{2_y} = N_s \cdot W_{\max,y}$$
- 7) se os deslocamentos transversais calculados forem iguais (a menos de uma certa tolerância) aos deslocamentos transversais calculados na iteração anterior, fim do processo iterativo → o pilar é estável;
- 8) se os deslocamentos transversais divergirem (não tenderem a um valor finito), fim do processo iterativo → o pilar não é estável;
- 9) volta-se ao passo 1.

³⁶ A materialização computacional deste item é a sub-rotina “VERIFIC_PPADRAO”, constante do Apêndice 1.

7.4. VERIFICAÇÃO DA ESTABILIDADE PELO MÉTODO GERAL

Conforme visto, o método do equilíbrio tem por objetivo calcular somente se o pilar é ou não estável, não informando nada sobre o carregamento máximo a que pode ser submetido o pilar. O método geral tem por objetivo calcular a carga máxima a que o pilar pode ser submetido; para tanto, calcula-se o diagrama carga–deslocamento do pilar.

O diagrama carga–deslocamento pode ser confeccionado adotando-se um parâmetro k que represente os esforços solicitantes do pilar em cada iteração e escolhendo-se os deslocamentos em uma seção para a aferição da estabilidade da configuração de equilíbrio (fig. 7.8). Portanto, pode-se calcular o diagrama carga–deslocamento de duas maneiras:

- a) variando-se as forças normais nas seções do pilar e mantendo-se constantes os momentos fletores de primeira ordem nas seções do pilar:

$$N_{s,i} = k \cdot N_{s,i,referencia}, \text{ onde} \quad (7.20)$$

$N_{s,i}$ = força normal solicitante na seção i na iteração em cálculo

$N_{s,i,referencia}$ = força normal solicitante de referência na seção i (por exemplo, os esforços solicitantes de serviço)

k = parâmetro que representa os esforços solicitantes do pilar na iteração em cálculo

- b) variando-se os momentos fletores de primeira ordem nas seções do pilar e mantendo-se constantes as forças normais nas seções do pilar:

$$\begin{aligned} M_{sx,i} &= k \cdot M_{sx,i,referencia} \\ M_{sy,i} &= k \cdot M_{sy,i,referencia} \end{aligned} \quad (7.21)$$

$M_{sx,i}$ e $M_{sy,i}$ = momentos fletores solicitantes na seção i na iteração em cálculo

$M_{sx,i,referencia}$ e $M_{sy,i,referencia}$ = momentos fletores solicitantes de referência na seção i (por exemplo, os esforços solicitantes de serviço)

k = parâmetro que representa os esforços solicitantes do pilar na iteração em cálculo.

Observe que o cálculo dos deslocamentos transversais em cada iteração pode ser feito através do método do equilíbrio, tanto com o processo geral quanto com o processo do pilar padrão.

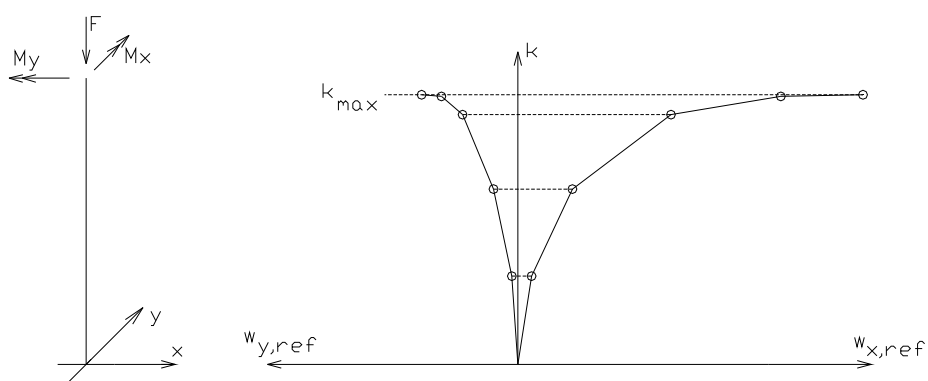


FIGURA 7.8 - Diagrama carga-deslocamento.

Apresenta-se a seguir o algoritmo proposto para a verificação da estabilidade de pilares pelo método geral:

- 1) define-se o passo com que k será incrementado (Δk) em cada iteração (como sugestão, $\Delta k=0,1$);
- 2) $k=0$;
- 3) com k , calcula-se os esforços solicitantes em todas as seções do pilar \rightarrow eq. 7.20 ou 7.21, conforme for o caso;
- 4) cálc. o deslocamento de referência e se o pilar rompe ou não: método do equilíbrio, item 7.2.5 ou 7.3.2, conforme esteja se utilizando o processo exato ou o processo do pilar padrão;
- 5) se o pilar não rompe:
 - $k=k+\Delta k$;
 - volta-se ao passo 3;
- 6) se o pilar rompe, fim do processo:
 - $k_{\max}=k-\Delta k$;
 - cálc. dos esforços solicitantes máximos a que o pilar pode suportar \rightarrow eq. 7.20 ou 7.21, conforme o caso;

7.5. DIMENSIONAMENTO DA ARMADURA DE PILARES ESBELTOS

O problema a ser resolvido neste item é: dado um pilar totalmente definido (a menos da área das barras de aço) e os esforços solicitantes, calcular a menor área de cada barra da armadura (A_{si}) que propicia que o pilar resista aos esforços solicitantes dados sem romper³⁷. Supõe-se que todas as barras da armadura têm a mesma área. Esquemáticamente, tem-se:

$$\left\{ \begin{array}{l} \text{pilar (a menos de } A_{si} \text{)} \\ N_s, M_{sx}, M_{sy} \end{array} \right\} \Rightarrow \{A_{si}, \text{ tal que o pilar não rompa}\}$$

A resposta do problema exige um cálculo por tentativas:

- 1) arbitra-se A_{si}
- 2) verifica-se se o pilar rompe ou não → método do equilíbrio, item 7.2.5 ou 7.3.2, conforme esteja se utilizando o processo exato ou o processo do pilar padrão;
- 3) se a seção rompe → volta-se ao passo 1

Para otimizar este processo, pode-se lançar mão de um método numérico iterativo de cálculo de zeros de funções³⁸. No caso, o Método da Bissecção adapta-se perfeitamente. Conforme explica-se no Anexo A, o Método da Bissecção necessita, para começar o processo iterativo, que se determine um intervalo inicial $[a,b]$ que contenha internamente a raiz procurada ($a \leq A_{si} \leq b$). Para se encontrar esse intervalo, pode-se lançar mão do Método dos Intervalos Encaixantes.

Apresenta-se a seguir o algoritmo proposto para o dimensionamento da armadura de pilares:

- 1) $a=0$ ('a' e 'b' representam a área de cada barra de aço)
- 2) $A_{si}=a$ → calc. se o pilar rompe ou não: método do equilíbrio, item 7.2.5 ou 7.3.2, conforme esteja se utilizando o processo exato ou o processo do pilar padrão
- 3) se o pilar rompeu, $ya=-1$; senão, $ya=+1$
- 4) se o pilar não rompeu: fim do processo
- 5) $b=1$

³⁷ A materialização computacional deste item é a sub-rotina "DIM_EXATO" ou "DIM_PPADRAO", conforme o processo utilizado para a verificação da estabilidade, constante do Apêndice 1.

³⁸ Ver Anexo A ou RUGGIERO & LOPES(1988).

- 6) $A_{si}=b \rightarrow$ cálc. se o pilar rompe ou não: método do equilíbrio, item 7.2.5 ou 7.3.2, conforme esteja se utilizando o processo exato ou o processo do pilar padrão
- 7) se o pilar rompeu, $y_b=-1$; senão, $y_b=+1$
- 8) se $y_a \cdot y_b > 0$: o intervalo $[a,b]$ encontrado ainda não contém raiz \rightarrow Método dos Intervalos Encaixantes
- 8.1) $a=b$; $y_a=y_b$
- 8.2) $b=b+1$
- 8.3) $A_{si}=b \rightarrow$ cálc. se o pilar rompe ou não: método do equilíbrio, item 7.2.5 ou 7.3.2, conforme esteja se utilizando o processo exato ou o processo do pilar padrão
- 8.4) se o pilar rompeu, $y_b=-1$; senão, $y_b=+1$
- 8.5) volta-se ao passo 8
- 9) cálc. da raiz: Método da Bisseção
- 9.1) $M=y_a$
- 9.2) $x=(a+b)/2$
- 9.3) $A_{si}=x \rightarrow$ cálc. se o pilar rompe ou não: método do equilíbrio, item 7.2.5 ou 7.3.2, conforme esteja se utilizando o processo exato ou o processo do pilar padrão
- 9.4) se o pilar rompeu, $y_x=-1$; senão, $y_x=+1$
- 9.5) se $M \cdot y_x > 0$: $a=x$; $y_a=y_x$;
senão: $b=x$; $y_b=y_x$
- 9.6) se $(b-a) > \text{TOLERÂNCIA}$: volta-se ao passo 9.2.

CAPÍTULO 8 - APLICAÇÕES NUMÉRICAS

Apresentam-se, neste capítulo, exemplos de verificação e dimensionamento de seções isoladas e pilares esbeltos efetuados com base nos algoritmos desenvolvidos neste trabalho, através do programa computacional FLEXOR (apêndice 1). Os exemplos foram escolhidos visando demonstrar a eficácia do programa FLEXOR e a comparação dos resultados obtidos com os resultados de trabalhos correlatos a este.

8.1. EXEMPLO 1

Para a seção da figura 8.1, verificar a segurança para os esforços solicitantes de: $N_d=1000\text{kN}$; $M_{dx}=-10200\text{kN.cm}$; $M_{dy}=4350\text{kN.cm}$ (referidos ao CG da seção bruta de concreto).

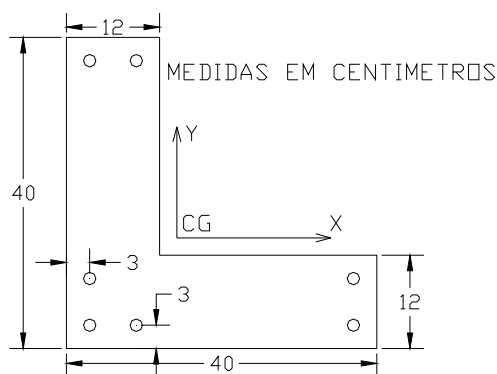


FIGURA 8.1 - Seção dos exemplos 1, 2 e 3.

Dados: $f_c = 1,1333\text{kN/cm}^2$

$f_y = 43,478\text{kN/cm}^2$

$E_s = 20000\text{kN/cm}^2$

classe do aço = "A"

$A_{si} = 2\text{cm}^2$ (área de cada barra de aço)

A direção do momento solicitante é: $\beta_s = \arctan(M_{dy}/M_{dx}) = 156,9^\circ$. Para este β_s , o programa FLEXOR forneceu os seguintes momentos resistentes máximos: $M_{rx}=-10285,99\text{kN.cm}$; $M_{ry}=4386,69\text{kN.cm}$ (direção da linha neutra: $\alpha = 92,96^\circ$). Portanto, a

seção suporta os esforços solicitantes dados. O resultado coincide com o apresentado por SANTOS (1994).

8.2. EXEMPLO 2

Para a seção da figura 8.1, dimensionar a armadura para que a seção suporte os esforços solicitantes de: $N_d=1000\text{kN}$; $M_{dx}=0\text{kN.cm}$; $M_{dy}=0\text{kN.cm}$ (observe que não se trata de compressão uniforme, pois a seção não é duplamente simétrica).

O programa FLEXOR forneceu a seguinte área de armadura: $A_{si} = 0,2717\text{cm}^2$ (área de cada barra de aço). O resultado difere em 0,6% em relação ao apresentado por SANTOS (1994). Isto se deve, provavelmente, à utilização de tolerâncias diferentes nos dois programas.

8.3. EXEMPLO 3

Para a seção da figura 8.1 ($A_{si} = 2\text{cm}^2$), confeccionar o diagrama de interação no ELU para força normal resistente $N_r = 1000\text{kN}$.

Os dados fornecidos pelo programa FLEXOR permitiram confeccionar o diagrama a seguir. Tal diagrama coincide com o apresentado por SANTOS (1994).

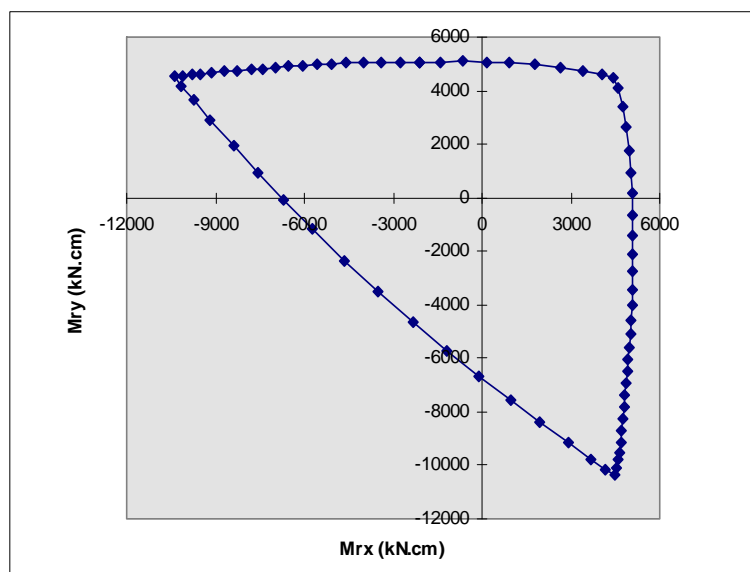


FIGURA 8.2 - Diagrama de interação do exemplo 3.

8.4. EXEMPLO 4

Dimensionar a armadura (suposta constante ao longo do comprimento) do pilar apresentado na figura 8.3 pelo processo exato (discretizar o pilar com 8 trechos iguais, 9 seções). Supor que o carregamento seja de curta duração. As cargas dadas na figura 8.3 já são valores de cálculo.

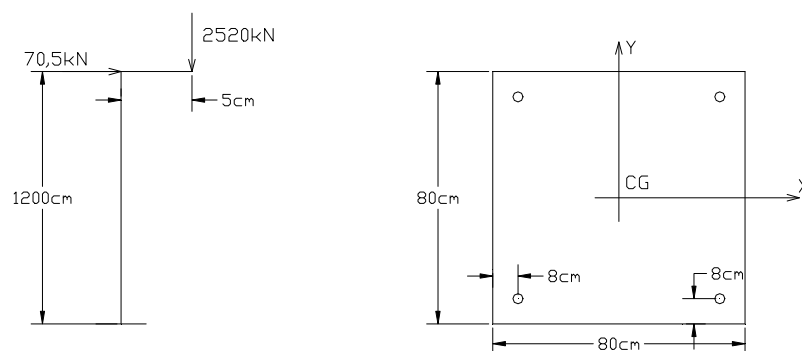


FIGURA 8.3 - Pilar do exemplo 4.

Dados: $f_c = 1,275 \text{ kN/cm}^2$

$f_y = 43,478 \text{ kN/cm}^2$

$E_s = 21000 \text{ kN/cm}^2$

classe do aço = "A"

O programa FLEXOR forneceu a seguinte área de armadura: $A_{si} = 19,38\text{cm}^2$ (área de cada barra de aço). O resultado difere em 3% em relação ao apresentado por PAULA (1988). Isto se deve, provavelmente, à utilização de tolerâncias diferentes nos dois programas. A flecha no topo do pilar foi de 29,17cm.

Resolver novamente o exemplo supondo-se que todo o carregamento seja de longa duração. Utilizar $\phi=2$ (função de fluência).

O programa FLEXOR forneceu a seguinte área de armadura: $A_{si} = 25,12\text{cm}^2$ (aumento de 29% em relação ao cálculo com $\phi=0$). A flecha no topo do pilar foi de 39,98cm (aumento de 37% em relação ao cálculo com $\phi=0$).

8.5. EXEMPLO 5

Dimensionar a armadura (suposta constante ao longo do comprimento) do pilar apresentado na figura 8.4 pelo processo exato (discretizar o pilar em 5 trechos iguais, 6 seções). Supor que o carregamento seja de curta duração.

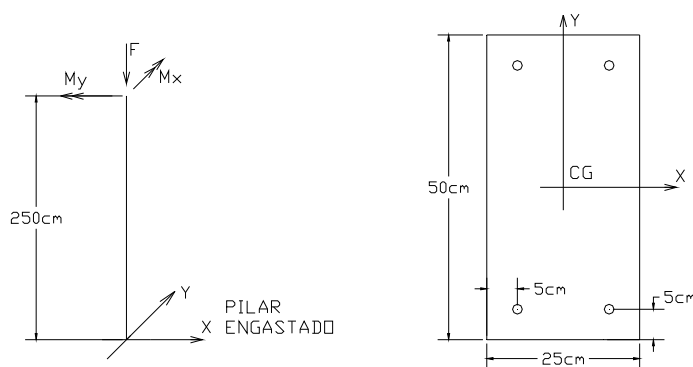


FIGURA 8.4 - Pilar do exemplo 5.

Dados: $f_c = 1,062\text{kN/cm}^2$

$f_y = 50\text{kN/cm}^2$

$E_s = 21000\text{kN/cm}^2$

classe do aço = "A"

cargas aplicadas no topo do pilar (valores de cálculo):

$F = 531,3\text{kN}$

$M_x = 2585,8\text{kN.cm}$

$$M_y = 5171,76\text{kN.cm}$$

O programa FLEXOR forneceu a seguinte área de armadura: $A_{si} = 1,97\text{cm}^2$ (área de cada barra de aço). O resultado difere em 1,5% em relação ao apresentado por MARCOTTI (1984). Isto se deve, provavelmente, ao fato do carregamento utilizado neste exemplo ter sido obtido a partir do ábaco elaborado por aquele autor para este pilar. Flecha no topo do pilar: $w_x = 4,6\text{cm}$ e $w_y = 1,27\text{cm}$.

Resolver novamente o exemplo supondo-se que todo o carregamento seja de longa duração. Utilizar $\phi=2$ (função de fluência).

O programa FLEXOR forneceu a seguinte área de armadura: $A_{si} = 3,3\text{cm}^2$ (aumento de 67% em relação ao cálculo com $\phi=0$). Flecha no topo do pilar: $w_x = 7,07\text{cm}$ (aumento de 53% em relação ao cálculo com $\phi=0$) e $w_y = 1,48\text{cm}$ (aumento de 16% em relação ao cálculo com $\phi=0$).

8.6. EXEMPLO 6

Dimensionar a armadura (suposta constante ao longo do comprimento) do pilar do exemplo 5 (fig. 8.4) pelo processo do pilar padrão. Supor que o carregamento seja de curta duração.

Cargas aplicadas no topo do pilar (valores de cálculo):

$$F = 531,3\text{kN}$$

$$M_x = 2664,2\text{kN.cm}$$

$$M_y = 5328,4\text{kN.cm}$$

O programa FLEXOR forneceu a seguinte área de armadura: $A_{si} = 1,97\text{cm}^2$ (área de cada barra de aço). O resultado difere em 1,5% em relação ao apresentado por MARCOTTI (1984). Isto se deve, provavelmente, ao fato do carregamento utilizado neste exemplo ter sido obtido a partir do ábaco elaborado por aquele autor para este pilar. Flecha no topo do pilar: $w_x = 4,36\text{ cm}$ e $w_y = 1,15\text{cm}$.

Resolver novamente o exemplo supondo-se que todo o carregamento seja de longa duração. Utilizar $\phi=2$ (função de fluência).

O programa FLEXOR forneceu a seguinte área de armadura: $A_{si} = 3,23\text{cm}^2$ (aumento de 64% em relação ao cálculo com $\phi=0$). Flecha no topo do pilar: $w_x = 6,68\text{cm}$ (aumento de 52% em relação ao cálculo com $\phi=0$) e $w_y = 1,38\text{cm}$ (aumento de 20% em relação ao cálculo com $\phi=0$).

CAPÍTULO 9 - CONCLUSÕES E COMENTÁRIOS

Pode-se concluir que os procedimentos e algoritmos apresentados neste trabalho tem grande aplicação na área de concreto armado, particularmente no dimensionamento e verificação de pilares esbeltos de concreto armado com seção transversal qualquer solicitados por flexão composta oblíqua, levando-se em consideração as não-linearidades física e geométrica. Para tal, apresentou-se o método geral e o método do equilíbrio, tanto com o processo exato quanto com o processo do pilar padrão. Também pode-se concluir que os procedimentos aqui expostos funcionam satisfatoriamente com qualquer situação de carregamento e seção transversal, ao contrário da grande maioria dos trabalhos correlatos a este. As conclusões de porque isto ocorre encontram-se no item 5.5.

Os procedimentos apresentados neste trabalho foram convertidos em sub-rotinas computacionais (sistema FLEXOR) que automatizam os métodos descritos acima, possibilitando o dimensionamento e verificação tanto de pilares esbeltos quanto de seções isoladas no ELU de ruptura. Tais sub-rotinas mostraram-se uma ferramenta poderosa à disposição tanto de projetistas de estruturas quanto de pesquisadores, para enfrentarem problemas que até agora só podiam ser resolvidos com métodos aproximados.

O sistema FLEXOR é especialmente útil no dimensionamento e verificação de pilares de grande esbeltez (onde a utilização do processo exato e a consideração da fluência são obrigatórias), pilares de seção variável, de forma irregular (seção L, Z, T, U, etc.), que fazem parte da estrutura de contraventamento de edifícios altos, etc.

Para pilares usuais de edifícios, a utilização do sistema FLEXOR possibilita economia no dimensionamento da armadura, pois os métodos aproximados facultados em normas são, geralmente, a favor da segurança, e, portanto, contra a economia.

Os pilares de pontes também podem ser estudados com auxílio das sub-rotinas aqui expostas, possibilitando a comparação do desempenho de diferentes concepções estruturais, viabilizando a redução de custos.

Apresenta interesse, também, a comparação de desempenho entre elementos estruturais de seção cheia e de seção vazada, possibilitando a escolha da solução que melhor resolve determinado problema.

Como continuidade deste trabalho, sugere-se a inclusão dos efeitos de flexo-torção (tensões normais oriundas de bimomentos), a ampliação do trabalho para a análise de pórticos espaciais com não-linearidades física e geométrica, a inclusão da teoria de probabilidades na análise de pilares, a generalização dos procedimentos para a análise de elementos confeccionados com concreto de alto desempenho ou protendido, a consideração de deformações iniciais diferentes dos materiais (como ocorre em casos de reforço de estruturas de concreto ou nas estruturas mistas concreto-aço), etc.

ANEXO A - MÉTODOS NUMÉRICOS PARA O CÁLCULO DE ZEROS DE FUNÇÕES

A.1. INTRODUÇÃO

Este anexo consiste em uma transcrição adaptada e resumida do capítulo 2 de RUGGIERO & LOPES (1988). Muitos dos itens deste anexo não passam de simples cópia daquela bibliografia. Para informações complementares e demonstrações dos métodos, recomenda-se consultar aquela bibliografia.

Nas mais diversas áreas das ciências exatas ocorrem, freqüentemente, situações que envolvem a resolução de uma equação do tipo $f(x)=0$. O objetivo deste anexo é o estudo de métodos numéricos para a resolução de equações deste tipo.

Um número real a é um zero da função $f(x)$ ou uma raiz da equação $f(x)=0$ se $f(a)=0$.

Em alguns casos, por exemplo de equações polinomiais, os valores x que anulam $f(x)$ podem ser reais ou complexos. Neste anexo, estaremos interessados somente nos zeros reais de $f(x)$, os quais denotaremos por ξ .

Graficamente, os zeros reais são representados pelas abscissas dos pontos onde uma curva intercepta o eixo Ox , fig. A.1.

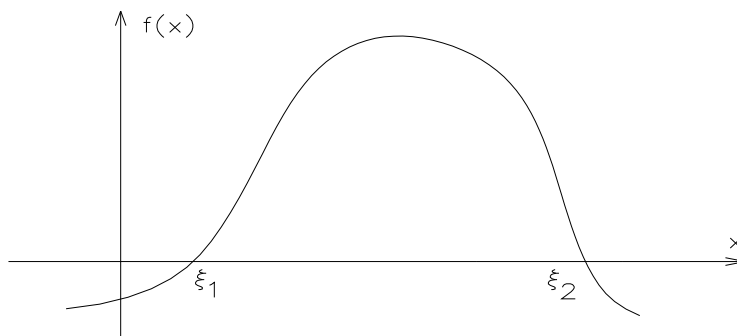


FIGURA A.1 - Zeros reais.

Sabemos que para algumas equações, como por exemplo as equações polinomiais de segundo grau, existem fórmulas explícitas que dão as raízes em função dos coeficientes da equação. No entanto, no caso de polinômios de grau mais alto e, pior ainda, no caso de funções mais complicadas, é praticamente impossível se achar os zeros exatamente. Por isso temos de nos contentar em encontrar apenas aproximações para esses zeros; mas isto não é uma limitação muito séria, pois com os métodos apresentados neste anexo, conseguimos, a menos de limitações numéricas, encontrar os zeros de uma função com qualquer precisão prefixada.

A idéia central destes métodos é partir de uma aproximação inicial para a raiz e em seguida *refinar* essa aproximação através de um *Processo Iterativo*.

Por isso os métodos constam de duas fases:

FASE I: Localização ou Isolamento das Raízes, que consiste em obter um intervalo que contém a raiz ξ ;

FASE II: Refinamento, que consiste em, escolhidas aproximações iniciais no intervalo encontrado na Fase I, melhorá-las sucessivamente até se obter uma aproximação para a raiz dentro de uma precisão ε prefixada.

A.2. FASE I - ISOLAMENTO DAS RAÍZES

Nesta fase é feita a análise teórica e gráfica da função $f(x)$.

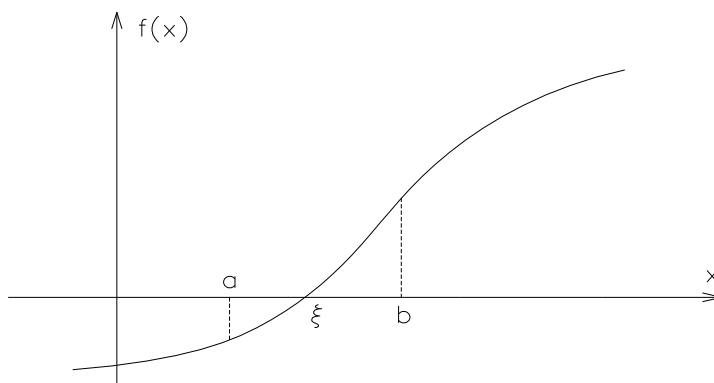
É importante ressaltar que o sucesso da Fase II depende fortemente da precisão desta análise.

Na análise teórica usamos freqüentemente o teorema:

TEOREMA 1

Seja $f(x)$ uma função contínua num intervalo $[a,b]$.

Se $f(a) \cdot f(b) < 0$ então existe pelo menos um ponto $x = \xi$ entre a e b que é zero de $f(x)$.

GRAFICAMENTE:FIGURA A.2 - Intervalo $[a,b]$ com raiz.

OBSERVAÇÃO: Sob as hipóteses do teorema anterior, se $f'(x)$ existir e preservar sinal em (a,b) , então este intervalo contém um único zero de $f(x)$.

Uma forma de se isolar as raízes de $f(x)$ usando os resultados anteriores é tabelar $f(x)$ para vários valores de x e analisar as mudanças de sinal de $f(x)$ e o sinal da derivada nos intervalos em que $f(x)$ mudou de sinal.

A.2.1. MÉTODO DOS INTERVALOS ENCAIXANTES

O Método dos Intervalos Encaixantes, neste trabalho, é um método utilizado para encontrar um intervalo $[a,b]$ que contenha em seu interior a raiz procurada. Consiste em se arbitrar valores seqüenciais de x até que $f(x_i) \cdot f(x_{i-1}) < 0$ (fig. A.3).

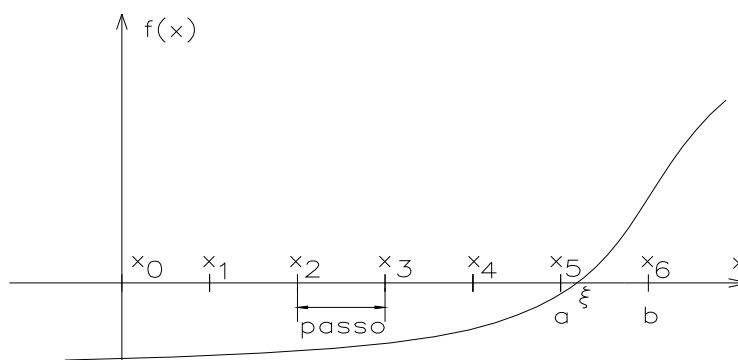


FIGURA A.3 - Método dos Intervalos Encaixantes.

ALGORITMO:

- 1) $a=0$
- 2) cálc. $f(a)$
- 3) $b=a+\text{passo}$
- 4) cálc. $f(b)$
- 5) se $f(a) \cdot f(b) > 0$:
 - 5.1) $a=b$
 - 5.2) $f(a)=f(b)$
 - 5.3) volta-se ao item 3

Observe que o tamanho do intervalo desejado depende do valor do *passo* utilizado no incremento de x . Se se deseja como resultado um intervalo pequeno, o processo pode implicar em um grande número de iterações. Para minimizar este problema, pode-se começar o processo com um *passo* grande e, quando o processo encontrar um intervalo com raiz, recomeçar o processo dentro do intervalo encontrado mas com um *passo* menor (por exemplo, um décimo do *passo* inicial).

A.3. FASE II - REFINAMENTO

Estudaremos neste item vários métodos numéricos de refinamento de raiz.

A forma como se efetua o refinamento é que diferencia os métodos. Todos eles pertencem à classe dos métodos iterativos.

Um *método iterativo* consiste de uma seqüência de instruções que são executadas “passo a passo”, algumas das quais são repetidas em ciclos.

A execução de um ciclo recebe o nome de *iteração*. Cada iteração utiliza resultados das iterações anteriores e efetua determinados testes que permitem verificar se foi atingido um resultado “próximo o suficiente” do resultado esperado.

Observamos que os métodos iterativos fornecem apenas uma aproximação para a solução, enquanto os métodos diretos, teoricamente, obtêm a solução exata da equação.

Os métodos iterativos para o refinamento da aproximação inicial para a raiz exata podem ser colocados num algoritmo:

- 1) cálculos iniciais
- 2) $k=1$
- 3) calcular a nova aproximação
- 4) se essa aproximação está “próxima o suficiente” da raiz exata \rightarrow fim do processo
- 5) $k=k+1$
- 6) volta-se ao passo 3

A.3.1. CRITÉRIOS DE PARADA

Pelo algoritmo, verifica-se que todos os métodos iterativos para obter zeros de função efetuam um teste do tipo: x_k está suficientemente próximo da raiz exata?

Que tipo de teste efetuar para se verificar se x_k está “suficientemente próximo” da raiz exata? Para isto é preciso entender o significado de “raiz aproximada”.

Existem duas interpretações para “raiz aproximada” que nem sempre levam ao mesmo resultado:

x_{raiz} é raiz aproximada com precisão ε se:

$$\text{I) } |x_{\text{raiz}} - \xi| < \varepsilon$$

$$\text{II) } |f(x_{\text{raiz}})| < \varepsilon$$

Como efetuar o teste I se não se conhece ξ ? Uma forma é reduzir o intervalo que contém a raiz a cada iteração. Ao se conseguir um intervalo $[a,b]$ tal que:

$$\left. \begin{array}{l} \xi \in [a, b] \\ e \\ b - a < \varepsilon \end{array} \right\} \text{então, } \forall x \in [a, b] \text{ temos } |x - \xi| < \varepsilon. \text{ Portanto, } \forall x \in [a, b] \text{ pode ser tomado}$$

como x_{raiz} .

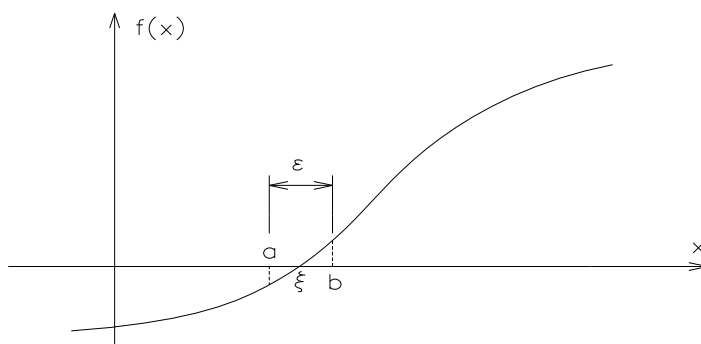


FIGURA A.4 - Intervalo $[a,b]$ que atende à precisão ε .

Observamos que dependendo da ordem de grandeza dos números envolvidos é aconselhável usar os testes do *erro relativo* que consistem em:

- I) reduzir o intervalo $[a,b]$ até que se tenha $\frac{b-a}{|b|} < \varepsilon$ ou $\frac{b-a}{|a|} < \varepsilon$
- II) considerar x_{raiz} como a aproximação de ξ se $\frac{|f(x_{\text{raiz}})|}{L} < \varepsilon$ onde $L = |f'(x)|$ para algum x escolhido numa vizinhança de ξ .

Nem sempre é possível ter ambas as exigências satisfeitas simultaneamente, conforme mostram as figuras seguintes:

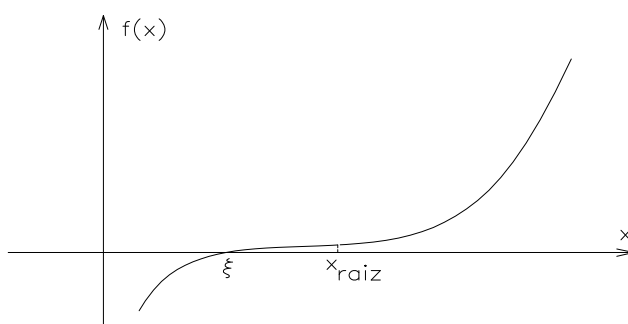


FIGURA A.5 - Exemplo onde se tem $|f(x_{\text{raiz}})| < \varepsilon$ mas $|x_{\text{raiz}} - \xi| \gg \varepsilon$.

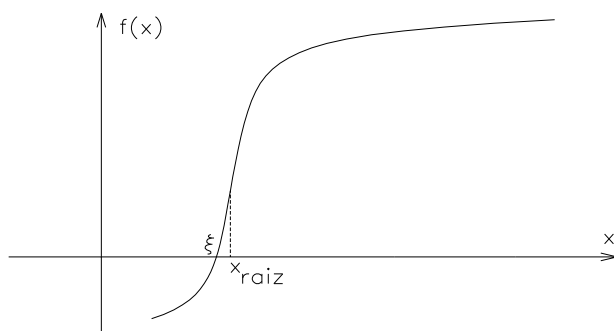


FIGURA A.6 - Exemplo onde se tem $|x_{raiz} - \xi| < \varepsilon$ mas $|f(x_{raiz})| \gg \varepsilon$.

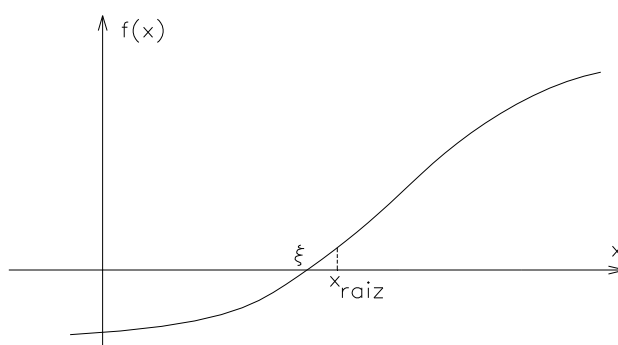


FIGURA A.7 - Exemplo onde se tem $|f(x_{raiz})| < \varepsilon$ e $|x_{raiz} - \xi| < \varepsilon$.

Os métodos numéricos são, em geral, desenvolvidos de forma a satisfazer um dos critérios.

Em programas computacionais, além do teste de parada usado para cada método, deve-se ter o cuidado de estipular um número máximo de iterações para se evitar que o programa entre em “looping” devido a erros no próprio programa ou à inadequação do método usado para o problema em questão.

A.3.2. MÉTODOS ITERATIVOS PARA SE OBTER ZEROS REAIS DE FUNÇÕES

A.3.2.1. MÉTODO DA BISSECCÃO

Seja a função $f(x)$ contínua no intervalo $[a,b]$ e tal que $f(a) \cdot f(b) < 0$.

Vamos supor, para simplificar, que o intervalo $[a,b]$ contém uma única raiz da equação $f(x)=0$.

O objetivo do método é reduzir a amplitude do intervalo que contém a raiz até se atingir a precisão requerida: $(b-a) < \varepsilon$, usando para isto a sucessiva divisão de $[a,b]$ ao meio (fig. A.8).

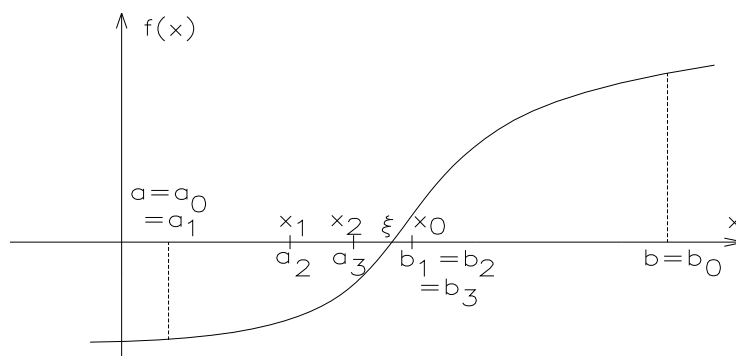


FIGURA A.8 - Exemplo do método da Bissecção.

ALGORITMO:

Seja $f(x)$ contínua em $[a,b]$ e tal que $f(a) \cdot f(b) < 0$.

- 1) dados iniciais:
 - a) intervalo inicial $[a,b]$
 - b) precisão ε_x
- 2) se $(b-a) < \varepsilon_x$, escolhe-se para x_{raiz} qualquer $x \in [a, b]$; fim do processo
- 3) $k=1$
- 4) $M=f(a)$
- 5) $x = \frac{a + b}{2}$
- 6) se $M \cdot f(x) > 0$, faz-se $a=x$; vai-se para o passo 8
- 7) $b=x$
- 8) se $(b-a) < \varepsilon_x$, escolhe-se para x_{raiz} qualquer $x \in [a, b]$; fim do processo
- 9) $k=k+1$; volta-se ao passo 5

Terminado o processo, teremos um intervalo $[a,b]$ que contém a raiz (e tal que $(b-a) < \varepsilon_x$) e uma aproximação x_{raiz} para a raiz exata.

CONVERGÊNCIA:

Se $f(x)$ é contínua no intervalo $[a,b]$ e $f(a) \cdot f(b) < 0$, então o método da Bisseção converge.

COMENTÁRIOS FINAIS:

- conforme demonstramos, o método converge sempre, ou seja, é sempre possível obter um intervalo que contém a raiz da equação em estudo, sendo que o comprimento deste intervalo final satisfaz a precisão requerida. Portanto, o método pode ser aplicado para obter a raiz de qualquer equação;
- as iterações não envolvem cálculos laboriosos;
- a convergência é muito lenta, pois se o intervalo inicial é tal que $b_0 - a_0 \gg \varepsilon$ e se ε for “muito pequeno” o número de iterações tende a ser muito grande.

A.3.2.2. MÉTODO DA POSIÇÃO FALSA

Seja a função $f(x)$ contínua no intervalo $[a,b]$ e tal que $f(a) \cdot f(b) < 0$.

Supor que o intervalo $[a,b]$ contém uma única raiz da equação $f(x)=0$.

Podemos esperar conseguir a raiz aproximada x_{raiz} usando as informações sobre os valores de $f(x)$ disponíveis a cada iteração.

No caso do método da Bisseção, x é simplesmente a média aritmética entre a e b :

$$x = \frac{a + b}{2}.$$

O método da Posição Falsa, ao invés de tomar a média aritmética, toma a média aritmética ponderada entre a e b com pesos $|f(b)|$ e $|f(a)|$, respectivamente:

$$x = \frac{a|f(b)| + b|f(a)|}{|f(b)| + |f(a)|} = \frac{a \cdot f(b) - b \cdot f(a)}{f(b) - f(a)} \text{ visto que } f(a) \text{ e } f(b) \text{ têm sinais opostos.}$$

Este valor de x é o ponto de interseção entre o eixo Ox e a reta $r(x)$ que passa por $(a,f(a))$ e $(b,f(b))$, conforme mostra a figura A.9.

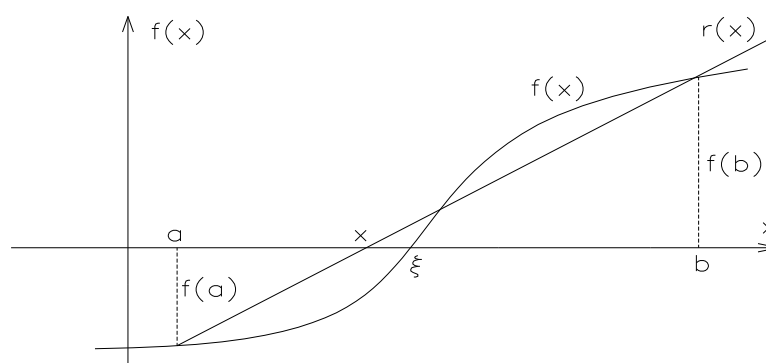


FIGURA A.9 - x no método da Posição Falsa.

ALGORITMO:

Seja $f(x)$ contínua em $[a,b]$ e tal que $f(a) \cdot f(b) < 0$.

- 1) dados iniciais:
 - a) intervalo inicial $[a,b]$
 - b) precisões ε_x e ε_y
- 2) se $(b-a) < \varepsilon_x$, escolhe-se para x_{raiz} qualquer $x \in [a, b]$; fim do processo
- 3) se $|f(a)| < \varepsilon_y$, faz-se $x_{\text{raiz}}=a$; fim do processo
- 4) se $|f(b)| < \varepsilon_y$, faz-se $x_{\text{raiz}}=b$; fim do processo
- 5) $k=1$
- 6) $M=f(a)$
- 7)
$$x = \frac{a \cdot f(b) - b \cdot f(a)}{f(b) - f(a)}$$
- 8) se $|f(x)| < \varepsilon_y$, faz-se $x_{\text{raiz}}=x$; fim do processo
- 9) se $M \cdot f(x) > 0$, faz-se $a=x$; vai-se para o passo 11
- 10) $b=x$
- 11) se $(b-a) < \varepsilon_x$, escolhe-se para x_{raiz} qualquer $x \in [a, b]$; fim do processo
- 12) $k=k+1$; volta-se ao passo 7

CONVERGÊNCIA:

Se $f(x)$ é contínua no intervalo $[a,b]$ e $f(a) \cdot f(b) < 0$, então o método da Posição Falsa converge.

COMENTÁRIOS FINAIS:

– em geral, o método da Posição Falsa obtém como raiz aproximada um ponto x_{raiz} no qual $|f(x_{\text{raiz}})| < \varepsilon_y$, sem que o intervalo $I=[a,b]$ seja “pequeno” o suficiente. Portanto, se for

exigido que os dois critérios de parada sejam satisfeitos simultaneamente, o método pode não convergir (computacionalmente isto significa exceder um número máximo de iterações);

– conforme vimos na observação anterior, muitas vezes no método da Posição Falsa o critério $b-a < \varepsilon$ pode não ser atingido.

A.3.2.3. MÉTODO DE NEWTON–RAPHSON

O método de Newton–Raphson consiste em, a partir de uma aproximação inicial x_0 , gerar uma seqüência $\{x_k\}$ de aproximações para ξ pela equação

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad (\text{A.1})$$

INTERPRETAÇÃO GEOMÉTRICA:

Dado x_k , o ponto x_{k+1} é obtido de tal forma que x_{k+1} é a abscissa do ponto de interseção entre o eixo Ox e a reta tangente à curva $f(x)$ em $(x_k, f(x_k))$, fig. A.10.

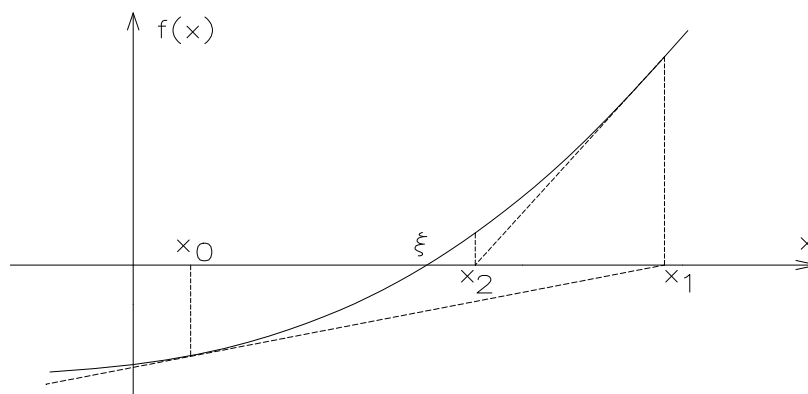


FIGURA A.10 - Exemplo do método de Newton–Raphson.

CONVERGÊNCIA:

Sejam $f(x)$, $f'(x)$ e $f''(x)$ contínuas num intervalo I que contém a raiz $x=\xi$ de $f(x)=0$. Supor que $f'(\xi) \neq 0$.

Então, existe um intervalo $\bar{I} \subset I$, contendo a raiz ξ , tal que, se $x_0 \in \bar{I}$, a seqüência $\{x_k\}$ gerada pela eq. A.1 convergirá para a raiz ξ .

Em geral, afirma-se que o método de Newton-Raphson converge desde que x_0 seja escolhido “suficientemente próximo” da raiz ξ .

Pelo denominador da eq. A.1, observa-se que não se deve escolher um número x_i tal que $f'(x_i)$ esteja próxima de zero.

Quanto a ordem de convergência, pode-se afirmar que, em geral, o método de Newton-Raphson tem convergência muito rápida.

ALGORITMO:

Seja a equação $f(x)=0$. Supor que as hipóteses dadas acima (item CONVERGÊNCIA) são satisfeitas.

- 1) dados iniciais:
 - a) x_0 : aproximação inicial
 - b) ε_x e ε_y : precisões
- 2) se $|f(x_0)| < \varepsilon_y$, faz-se $x_{\text{raiz}}=x_0$; fim do processo
- 3) $k=1$
- 4) $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$
- 5) se $|f(x_1)| < \varepsilon_y$, faz-se $x_{\text{raiz}}=x_1$; fim do processo
- 6) se $|x_1 - x_0| < \varepsilon_x$, faz-se $x_{\text{raiz}}=x_1$; fim do processo
- 7) $x_0=x_1$
- 8) $k=k+1$; volta-se ao passo 4

A.3.2.4. MÉTODO DA SECANTE

Uma grande desvantagem do método de Newton-Raphson é a necessidade de se obter e calcular o valor numérico de $f'(x)$ a cada iteração.

Uma forma de se contornar este problema é substituir a derivada $f'(x_k)$ pelo quociente das diferenças:

$$f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} \quad (\text{A.2})$$

onde x_k e x_{k-1} são duas aproximações para a raiz.

Substituindo-se a eq. A.2 em A.1, vem

$$x_{k+1} = x_k - \frac{f(x_k)}{f(x_k) - f(x_{k-1})} \cdot (x_k - x_{k-1}) \quad (\text{A.3})$$

OBSERVAÇÃO:

São necessárias duas aproximações para se inicializar o método.

INTERPRETAÇÃO GEOMÉTRICA:

A partir de duas aproximações x_{k-1} e x_k , o ponto x_{k+1} é obtido como sendo a abscissa do ponto de interseção do eixo Ox e da reta secante que passa por $(x_{k-1}, f(x_{k-1}))$ e $(x_k, f(x_k))$:

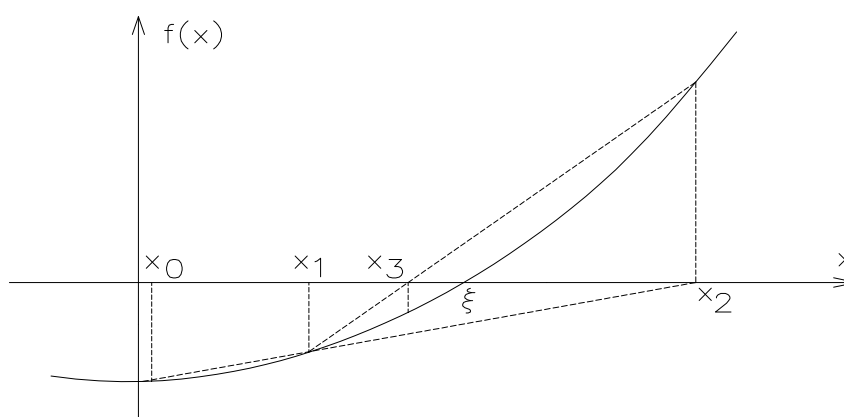


FIGURA A.11 - Exemplo do método da Secante.

ALGORITMO:

Seja a equação $f(x)=0$.

1) dados iniciais:

a) x_0 e x_1 : aproximações iniciais

b) ε_x e ε_y : precisões

2) se $|f(x_0)| < \varepsilon_y$, faz-se $x_{\text{raiz}}=x_0$; fim do processo

3) se $|f(x_1)| < \varepsilon_y$, faz-se $x_{\text{raiz}}=x_1$; fim do processo

4) se $|x_1 - x_0| < \varepsilon_x$, faz-se $x_{\text{raiz}}=x_1$; fim do processo

5) $k=1$

$$6) \quad x_2 = x_1 - \frac{f(x_1)}{f(x_1) - f(x_0)} \cdot (x_1 - x_0)$$

7) se $|f(x_2)| < \varepsilon_y$, faz-se $x_{\text{raiz}}=x_2$; fim do processo

8) se $|x_2 - x_1| < \varepsilon_x$, faz-se $x_{\text{raiz}}=x_2$; fim do processo

9) $x_0=x_1$; $x_1=x_2$

10) $k=k+1$; volta-se ao passo 6

COMENTÁRIOS FINAIS:

– visto que o método da Secante é uma aproximação do método de Newton-Raphson, as condições para a convergência do método são praticamente as mesmas; acrescente-se ainda que o método pode divergir se $f(x_k) \approx f(x_{k-1})$;

– a ordem de convergência do método da Secante é alta, mas menor que a do método de Newton-Raphson.

A.4. COMPARAÇÃO DOS MÉTODOS

A comparação entre os métodos deve levar em conta vários critérios entre os quais: garantias de convergência; rapidez de convergência; esforço computacional.

Conforme constatamos no estudo teórico, os métodos da Bisseção e Posição Falsa têm convergência garantida desde que a função seja contínua num intervalo $[a,b]$ tal que $f(a) \cdot f(b) < 0$. Já os métodos de Newton-Raphson e Secante têm condições mais restritivas de convergência. Por outro lado, estes últimos têm convergência mais rápida que os dois primeiros.

O esforço computacional é medido através do número de operações efetuadas a cada iteração, da complexidade destas operações, do número de decisões lógicas, do número de avaliação de funções a cada iteração e do número total de iterações.

Tendo isto em mente, percebe-se que é difícil tirar conclusões gerais sobre a eficiência computacional de um método pois, por exemplo, o método da Bisseção é o que efetua cálculos mais simples por iteração enquanto que o de Newton-Raphson requer cálculos mais elaborados pois requer o cálculo da função e de sua derivada a cada iteração. No entanto, o número de iterações efetuadas com o método da Bisseção pode ser muito maior que o número de iterações efetuadas com o método de Newton-Raphson.

Considerando que o método ideal seria aquele em que: a convergência estivesse assegurada; a ordem de convergência fosse alta e os cálculos por iteração fossem simples, o método de Newton-Raphson é o mais indicado sempre que for fácil verificar as condições de

convergência e que o cálculo de $f'(x)$ não seja muito elaborado. Nos casos em que é trabalhoso obter $f'(x)$ ou ainda em que $f'(x)$ é muito complicada é aconselhável usar o método da Secante, desde que ele é o que converge mais rapidamente entre as outras opções.

Outro item importante na escolha do método é o critério de parada, pois, por exemplo, se o objetivo é reduzir o intervalo que contém a raiz, não se deve usar métodos como a Posição Falsa – que apesar de trabalhar com intervalos pode não atingir a precisão requerida – nem Secante ou Newton-Raphson que trabalham exclusivamente com aproximações x_k para a raiz exata.

Após estas considerações, podemos concluir que a escolha do método está diretamente relacionada com a equação que se quer resolver, no que diz respeito ao comportamento da função na região da raiz exata, às dificuldades com o cálculo de $f'(x)$, ao critério de parada, etc.

REFERÊNCIAS BIBLIOGRÁFICAS

- AAS-JAKOBSEN, A. (1964). *Biaxial eccentricities in the ultimate load design*. ACI Journal Proceedings, v.61, n.3, mar.
- AHMAD, S. H. ; WEERAKOON, S. L. (1995). *Model for behavior of slender reinforced concrete columns under biaxial bending*. ACI Structural Journal, v.92, n.2, mar.-apr.
- ALLEN, H. G. ; BULSON, P. S. (1980). *Background to buckling*. London, McGraw-Hill.
- ARAÚJO, J. M. (1984). *Dimensionamento de pilares esbeltos de concreto armado*. Porto Alegre. Dissertação (mestrado) - Escola de Engenharia Civil, Universidade Federal do Rio Grande do Sul.
- ARAÚJO, J. M. (1985). *Cálculo de pilares de concreto armado*. Porto Alegre, Editora da Furg.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. (1978). *NBR 6118 - Cálculo e execução de obras de concreto armado*. São Paulo.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. (1994). *Texto base para a revisão da NBR 6118*. São Paulo.
- AUFIERO, L. (1977). *Estabilidade de colunas isostáticas de concreto armado*. São Carlos. Dissertação (mestrado) - Escola de Engenharia de São Carlos, Universidade de São Paulo.
- BRESLER, B. (1960). *Design criteria for reinforced columns under axial load and biaxial bending*. ACI Journal Proceedings, v.32, n.5, nov.

- BUCHAIN, R. (1979). *Efeitos de segunda ordem e estado limite último de instabilidade em pilares de concreto armado*. São Paulo. Dissertação (mestrado) - Escola Politécnica, Universidade de São Paulo.
- CABRÉ, F. M. (1972). *Cálculo de secciones de hormigon armado, sometidas a sollicitaciones normales, en el estado limite ultimo*. Madrid. Monografia n.304 del Instituto Eduardo Torroja.
- CAMPOS, N. B. F. (1994). *Contribuição ao estudo de pilares em concreto armado com seção qualquer submetidos à flexão composta oblíqua*. São Carlos. Dissertação (mestrado) - Escola de Engenharia de São Carlos, Universidade de São Paulo.
- COMITÉ EURO-INTERNATIONAL DU BÉTON. (1993). *CEB-FIP Model Code 1990 - Bulletin d'information No. 213/214*. Lausanne.
- DALL'ASTA, A.; DEZI, L. (1992). *Design of RC sections with generic shape under biaxial bending*. Journal of Structural Engineering, v.118, n.4, apr.
- DUNDAR, C.; SAHIN, B. (1993). *Arbitrarily shaped reinforced concrete members subject to biaxial bending and axial load*. Computers & Structures, v.49, n.4, nov.
- EL-METWALLY, S. E. E. (1994). *Method of segment length for instability analysis of reinforced concrete beam-column*. ACI Structural Journal, v.91, n.6, nov.-dec.
- FAVRE, R.; CHARIF, H. (1994). *Basic model and simplified calculations of deformations according to the CEB-FIP Model Code 1990*. ACI Structural Journal, v.91, n.2, mar.-apr.
- FORNI, F. E. B. (1980). *Flexão oblíqua composta no concreto armado*. São Paulo. Dissertação (mestrado) - Escola Politécnica, Universidade de São Paulo.
- FRANÇA, R. L. S. (1984). *Relações momento-curvatura em peças de concreto armado submetidas à flexão oblíqua composta*. São Paulo. Dissertação (mestrado) - Escola Politécnica, Universidade de São Paulo.

- FUSCO, P. B. (1981). *Estruturas de concreto armado, solicitações normais*. Rio de Janeiro, Guanabara Dois.
- HOFFMANN, J. R. (1979). *Pilares esbeltos de concreto armado - método exato*. Porto Alegre. Dissertação (mestrado) - Escola de Engenharia Civil, Universidade Federal do Rio Grande do Sul.
- HOROWITZ, B. (1989). *Design of columns subjected to biaxial bending*. ACI Structural Journal, v.86, n.6, nov.-dec.
- HOROWITZ, B. (1995). *Range and null space decomposition applied to analysis of slender concrete columns*. Computers & Structures, v.57, n.6.
- HSU, C. T. T.; FURLONG, R. W.; LEE, D. D.; MIRZA, S. A. PRISHTINA, B.; RAMAMURTHY, L. N.; RANGAN, B. V.; SAADEGHVAZIRI, M. A. (1994). *Recent advances in biaxially loaded columns*. In: Proc. Struct Congr 94, Publ by ASCE. New York. p.450-454.
- HWEI, C. P.; KEONG, K. S. (1995). *Failure loads of slender reinforced concrete columns*. Proc. Instn Civ. Engrs Structs & Bids, n.110, nov.
- KARN, L. F.; MEYER, K. F. (1995). *Rectangular stress block for nonrectangular compression zone*. ACI Structural Journal, v.92, n.3, may-June.
- LEONHARDT, F.; MONNIG, E. (1977). *Construções de concreto. princípios básicos do dimensionamento de estruturas de concreto armado*. Rio de Janeiro, Interciência.
- MARCOTTI, P. (1984). *Instabilidade na flexão composta oblíqua de pilares de concreto armado*. São Paulo. Dissertação (mestrado) - Escola Politécnica, Universidade de São Paulo.
- MARINO, M. A. (1978). *Seções transversais de concreto armado sujeitas a solicitações normais*. São Paulo. Dissertação (mestrado) - Escola Politécnica, Universidade de São Paulo.

- MENDES NETO, F. (1991). *Estudo de pilares de concreto armado submetidos a flexão oblíqua composta*. São Paulo. Dissertação (mestrado) - Escola Politécnica, Universidade de São Paulo.
- MIRZA, S. A. (1994). *Progress in reinforced concrete column design: an introduction*. In: Proc. Struct Congr 94, Publ by ASCE. New York. p.429-431.
- MONTOYA, P. J.; MESEGUER, A. G.; MORAN, F. (1978). *Hormigon armado*. 9ª ed. Barcelona. Editora Gustavo Gilli.
- PABLO, J. A. (1980). *Calculo de soportes rectangulares de hormigón armado em teoria de segundo orden*. Barcelona, Editores Tecnicos Asociados.
- PAULA, J. A. (1988). *Algoritmos para o estudo de pilares esbeltos de concreto armado solicitados à flexão normal composta*. São Carlos. Dissertação (mestrado) - Escola de Engenharia de São Carlos, Universidade de São Paulo.
- PINHEIRO, L. M.; BARALDI, L. T.; POREM, M. E. (1994). *Concreto armado: ábacos para flexão oblíqua*. São Carlos. Escola de Engenharia de São Carlos, Universidade de São Paulo.
- RUGGIERO, M. A. G.; LOPES, V. L. R. (1988). *Cálculo numérico, aspectos teóricos e computacionais*. São Paulo, McGraw-Hill. Cap. 2: Zeros de Funções Reais; Cap. 6: Integração Numérica.
- SANTOS, L. M. (1981). *Cálculo de concreto armado segundo a nova NB-1 e o CEB*. Vol.2. São Paulo, Editora LMS.
- SANTOS, L. M. (1987). *Determinação da resultante Rcc e da sua posição em função do encurtamento máximo e da curvatura - (seções diversas)*. São Paulo. Publicação interna do Departamento de Estruturas e Fundações, Escola Politécnica, Universidade de São Paulo.
- SANTOS, L. M. (1987). *Estado limite último de instabilidade*. São Paulo. Monografia - Escola Politécnica, Universidade de São Paulo.

- SANTOS, L. M. (1994). *Sub-rotinas básicas do dimensionamento de concreto armado*. Vol.1. São Paulo, Editora Thot.
- SAUAIA, W. (1981). *Estudo comparativo dos processos de dimensionamento à flexão oblíqua composta com redução a duas flexões normais*. São Paulo. Dissertação (mestrado) - Escola Politécnica, Universidade de São Paulo.
- SINHA, S. N. (1996). *Direct method of design of rectangular column section*. Structural Engineering and Mechanics, v.4, n.2, mar.
- THEWALT, C. R.; STOJADINOVIC, B. (1994). *Stable reinforced-concrete section-analysis procedure*. Journal of Structural Engineering, v.120, n.10, oct.
- THOMPSON, P. J.; ZAK, M. L. (1994). *Computer analysis of reinforced concrete sections under biaxial bending and longitudinal load*. ACI Structural Journal, v.91, n.1, jan.-feb.
- VENTURINI, W. S. (1989). *Dimensionamento de peças retangulares de concreto armado solicitadas à flexão reta*. São Carlos. Escola de Engenharia de São Carlos, Universidade de São Paulo.
- VENTURINI, W. S.; BORTOLIN, A. A. (1992). *Dimensionamento de peças retangulares de concreto armado solicitadas à flexão oblíqua*. São Carlos. Escola de Engenharia de São Carlos, Universidade de São Paulo.
- WANG, G.; HSU, C. T. (1990). *Complete load-deformation behavior of biaxially loaded reinforced concrete columns*. Technical Report Structural Series No.90-2, Department of Civil and Environmental Engineering, New Jersey Institute of Technology, sept.
- WERNER, H. (1974). *Schiefe biegun polygonal umrandeter stahlbeton-querschnitte*. Revista Beton Und Stahlbetonbau, abr.
- YAU, C. Y.; CHAN, S. L.; SO, A. K. W. (1993). *Biaxial bending design of arbitrarily shaped reinforced concrete column*. ACI Structural Journal, v.90, n.3, may-june.

- YEN, J. Y. R. (1991). *Quasi-Newton method for reinforced-concrete column analysis and design*. Journal of Structural Engineering, v. 117, n.3, mar.
- ZAGHW, A. (1995). Design of R. C. *Column subjected to biaxial bending using the direct model inversion (DMI) method*. Computing in Civil Engineering, v.1.
- ZAK, M. L. (1993). *Computer analysis of reinforced concrete sections under biaxial bending and longitudinal load*. ACI Structural Journal, v.90, n.2, mar.-apr.
- ZIELINSKI, Z. A.; LONG, W.; TROITSKY, M. S. (1995). *Designing reinforced concrete short-tied column using the optimization technique*. ACI Structural Journal, v.92, n.5, sep.-oct.

APÊNDICE 1 - LISTAGEM DO SISTEMA FLEXOR

Com base nos procedimentos e algoritmos dados nos Capítulos anteriores, foi elaborado o SISTEMA FLEXOR. Este programa foi escrito em linguagem FORTRAN POWERSTATION, e pode ser utilizado em praticamente qualquer computador da linha IBM-PC com plataforma WINDOWS 3.x ou superior.

O SISTEMA FLEXOR consiste de um conjunto de sub-rotinas que executam:

- 1) Dimensionamento de pilares esbeltos pelo Processo Exato: sub-rotina DIM_EXATO
- 2) Dimensionamento de pilares esbeltos pelo Processo do Pilar Padrão: sub-rotina DIM_PPADRAO
- 3) Dimensionamento de seções isoladas no ELU de ruptura: sub-rotina DIM_ELU
- 4) Verificação de pilares esbeltos pelo Método do Equilíbrio com Processo Exato: sub-rotina VERIFIC_EXATO
- 5) Verificação de pilares esbeltos pelo Método do Equilíbrio com Processo do Pilar Padrão: sub-rotina VERIFIC_PPADRAO
- 6) Verificação de seções isoladas no ELU de ruptura: sub-rotina VERIFIC_ELU
- 7) Outros cálculos: utilizando-se convenientemente as demais sub-rotinas do SISTEMA FLEXOR, pode-se, por exemplo:
 - confeccionar diagramas Momento Fletor–Força Normal–Curvatura (diagramas $M-N-\theta$) de seções: utiliza-se iterativamente a sub-rotina PROCURA_ALFA
 - confeccionar diagramas de interação M_x-M_y-N de seções no ELU de ruptura: utiliza-se iterativamente a sub-rotina PROCURA_e_CG_TETA_ELU
 - verificar pilares esbeltos pelo Método Geral: utiliza-se iterativamente as sub-rotinas VERIFIC_EXATO ou VERIFIC_PPADRAO (veja o algoritmo no item 7.4).

Salienta-se que a listagem do SISTEMA FLEXOR contém todas as sub-rotinas necessárias ao cálculo de pilares esbeltos e seções isoladas; porém, não contém os programas principais para não alongar demais a listagem e por não fazerem parte do escopo deste trabalho. Tais programas são somente executam a entrada dos dados necessários à utilização da sub-rotina adequada, e executam a saída dos dados. Como todas as sub-rotinas contém

uma lista explicativa de todas as variáveis de entrada e saída, os programas principais podem ser facilmente confeccionados.

O SISTEMA FLEXOR foi confeccionado de forma a ser o mais independente possível das normas técnicas que tratam do concreto armado. Somente três sub-rotinas foram confeccionadas com base em dados normativos:

- 1) e_CGxTETA_ELU: confeccionada com base nos domínios de deformação definidos na norma NB1/78
- 2) TETA_eCG_ELU: confeccionada com base nos domínios de deformação definidos na norma NB1/78
- 3) TENSAO_ACO: confeccionada com base nos diagramas tensão-deformação definidos na norma NB1/78 para os aços classe A ou B.

Caso a norma em vigor não seja a NB1/78, deve-se alterar as três sub-rotinas acima para que se adequem à norma em vigor. Todas as demais sub-rotinas do SISTEMA FLEXOR não dependem das normas técnicas e, portanto, não precisam ser alteradas caso a norma em vigor não seja a NB1/78. Salienta-se, ainda, que o SISTEMA FLEXOR não faz verificações quanto à taxa mínima de armadura, distância mínima entre as barras de aço, dimensões mínimas da seção transversal, limite de flecha, índice de esbeltez máxima, etc.

Nas páginas seguintes, encontra-se a listagem do SISTEMA FLEXOR.

```

*=====DIM_EXATO=====
*Subrotina que, dado um pilar totalmente definido (menos a área das barras da
* armadura) e seus esforços solicitantes, calcula a menor área de aço capaz de
* tornar o pilar estável. O cálculo é feito por tentativas: arbitra-se taxas de
* armadura e utiliza-se a subrotina verific_exato para verificar se tal pilar
* rompe ou não. Para otimizar este processo, esta subrotina utiliza um método
* numérico iterativo de cálculo de zeros de funções: o Método da Bissecção
* (ver Ruggiero, M.A.G. e Lopes, V.L.R. - "Cálculo Numérico - aspectos teóricos
* e computacionais"). Este método foi escolhido por ser eficiente e por ter
* convergência garantida para o tipo de função em questão rompeu=f(Asi), ao
* contrário de outros métodos mais eficientes, como o de Newton-Raphson ou da
* Secante. Para o entendimento da subrotina, recomenda-se consultar a
* bibliografia indicada acima. O pilar deve estar referenciado num sistema de
* coordenadas Oxyz dextrógiro, com seções transversais no plano xy e eixo do
* pilar ao longo do eixo z; a subrotina verific_exato utiliza o Método do
* Equilíbrio com o Processo Exato (veja as características e restrições de tal
* método em FUSCO,P.B.(1981) "Estruturas de Concreto - Solicitações Normais").
*Unidades de entrada: Sistema de unidades qualquer, desde que consistente;
*TOLERANCIA_Asi: Tolerância com que se calculará Asi (veja a definição de Asi
* abaixo); o Asi calculado pode conter um erro a favor da segurança de, no
* máximo, tolerancia_asi; a unidade de entrada deve ter dimensão L2 (área);
*N_SECAO: Número de seções transversais do pilar; o pilar deve ser discretizado
* com seções numeradas de 1 a N_SECAO; N_SECAO>=2;
*Z_SECAO: Vetor (N_SECAOx1) que contém as coordenadas z de cada seção
* transversal; a seção 1 deve ter z=0, e as demais z>0; duas seções podem ter o
* mesmo z (caso de pilar com variação brusca de seção ou de esforços solicit.);
*VETOR_N_SOLIC, VETOR_Mlx_SOLIC, VETOR_Mly_SOLIC: Vetores (N_SECAOx1) que contém
* os esforços de primeira ordem que solicitam as seções transversais do pilar
* (valores de cálculo); convenção de sinais:
*     valores      -> força normal de compressão;
*     positivos:   -> momento definido por uma excentricidade positiva no eixo x;
*     -> momento definido por uma excentricidade positiva no eixo y;
* não há restrições quanto à esses valores, que podem ser positivos, negativos
* ou zero;
*VETOR_N_SOLIC_CONTRAV: No caso de o pilar em análise ser uma estrutura de
* contraventamento (núcleo, pilar parede, etc.), VETOR_N_SOLIC_CONTRAV é um
* vetor (N_SECAOx1) que contém as forças normais totais que agem na estrutura
* contraventada no nível de cada seção transversal do pilar em análise (valores
* de cálculo); convenção de sinal: positivo se for de compressão;
*TIPO_PILAR_X, TIPO_PILAR_Y: Número que determina o tipo de vinculação
* (condições de contorno) do pilar nas direções x e y:
* 1 --> pilar biapoiado
* 2 --> pilar rigidamente engastado na base (seção 1) e livre no topo
* 3 --> pilar elasticamente engastado na base (seção 1) e livre no topo;
*K_MOLA_X, K_MOLA_Y: No caso de pilar do tipo 3, são as constantes de mola
* [FL/RADIANO] nas direções x e y do engaste elástico; k>0;
*Nmax_POLIG_CONCR, Nmax_BARRAS_ACO: Número de nós da poligonal de concreto da
* seção transversal que tiver a poligonal com o maior número de nós; e número
* de barras de aço da seção transversal que tiver a maior quantidade de barras
* de aço; (são os maiores valores dos vetores descritos a seguir);
*VETOR_N_POLIG_CONCR, VETOR_N_BARRAS_ACO: Vetores (N_SECAOx1) que contém o
* número de nós da poligonal de concreto (incluindo o nó de chegada, que é o
* mesmo nó de saída) de cada seção transversal; e contém o número de barras de
* aço de cada seção transversal;
*MATRIZ_X_POLIG_CONCR, MATRIZ_Y_POLIG_CONCR: Matrizes (N_SECAO x
* Nmax_POLIG_CONCR) que contém as coordenadas dos nós da poligonal de concreto
* de cada seção transversal (as poligonais deve ter seus nós numerados no
* sentido anti-horário, com vazios no sentido horário; com o nó de chegada, que
* é o mesmo nó de saída); Exemplo: MATRIZ_X_POLIG_CONCR(3,8) = coordenada x do
* nó 8 da poligonal de concreto que representa a seção transversal 3;
*MATRIZ_X_BARRAS_ACO, MATRIZ_Y_BARRAS_ACO: Matrizes (N_SECAO x Nmax_BARRAS_ACO)
* que contém as coordenadas das barras de aço de cada seção transversal;
* Exemplo: MATRIZ_X_BARRAS_ACO(3,8) = coordenada x da barra de aço 8 da seção
* transversal 3;
*N_m: Número de curvas (polinômios de grau qualquer) que compoem o diagrama
* tensão-deformação adotado para o concreto; cada curva é definida por um
* somatório do tipo:  $\sigma_c = C_0.e^0 + C_1.e^1 + C_2.e^2 + \dots + C_m.e^m$ ; para o
* diagrama parábola-retângulo da NBL/78, temos N_m=2 (curva parabólica e curva
* reta constante);
*VETOR_m: Vetor que contém os graus dos N_m polinômios; para o parábola-retângulo
* da NBL/78, temos VETOR_m=(2,0) (m=2 -> parábola; m=0 -> reta constante);
*MATRIZ_C: Matriz que contém os coeficientes dos N_m polinômios: linha i contém
* os Ck (k=1 à m+1) coeficientes do polinômio i; para o parábola-retângulo da
* NBL/78, temos: MATRIZ_C = !0 1 -0.25! -> coef. da parábola

```

```

*          !1 0 0 ! -> coef. da reta constante;
*MATRIZ_e_LIMITES: Matriz que contém as deformações (em por mil) iniciais e
* finais onde age cada um dos N_m polinômios (deformações estas sem o efeito da
* fluência): linha i contém as deformações iniciais e finais onde age o
* polinômio i; OBS: colocar um número grande para a deformação final do último
* polinômio que compõe o diagrama tensão-deformação adotado para o concreto,
* não finalizando o diagrama no ELU; para o parábola-retângulo da NBl/78, temos:
* MATRIZ_e_LIMITES = !0 2 ! -> def. limites da parábola
*          !2 1000! -> def. limites da reta constante;
*fc: Tensão máxima de compressão à que o concreto pode estar sujeito, ou seja,
* é a tensão máxima do diagrama tensão-deformação adotado para o concreto
* (valor de cálculo). Não há consenso quanto ao valor a se utilizar. A NBl/78
* indica fc=0.85fcd (ELU), mas, para o estudo da estabilidade, alguns autores
* indicam fc=0.85fck.
*FI_FLUENCIA: Função de Fluência (Teoria Linear de Fluência);
*fy: Tensão de escoamento do aço (valor de cálculo);
*Es: Módulo de Elasticidade do aço (usualmente igual a 210000MPa);
*CLASSE_ACO: É a classe à que pertence o aço (A ou B);
*VETOR_Wx, VETOR_Wy: Saídas da subrotina: Vetores (N_SECAOx1) que contém os
* deslocamentos nas direções x e y de cada seção transversal;
*VETOR_M2x_SOLIC, VETOR_M2y_SOLIC: Saídas da subrotina: Vetores (N_SECAOx1) que
* contém os momentos de segunda ordem que solicitam as seções transversais do
* pilar (valores de cálculo); convenção de sinais:
* valores -> momento definido por uma excentricidade positiva no eixo x;
* positivos: -> momento definido por uma excentricidade positiva no eixo y;
*Asi: Área de uma barra de aço (cada seção tem um total de n_barras_aco);
* considera-se que todas as barras de aço de todas as seções tem a mesma área.
*-----
* SISTEMA FLEXOR - DIMENSIONAMENTO E VERIFICAÇÃO DE PILARES ESBELTOS DE
* CONCRETO ARMADO COM SEÇÃO QUALQUER SOLICITADOS POR FLEXÃO
* COMPOSTA OBLÍQUA
* DIAG. TENSÃOxDEFORM.DO CONCRETO: POLINÔMIOS DE GRAU QUALQUER
* CONSIDERAÇÃO DA FLUÊNCIA: TEORIA LINEAR DE FLUÊNCIA
* AÇOS BRASILEIROS DO TIPO 'A' OU 'B' COM QUALQUER fy OU Es
* Copyright(c) 1997, Ismael Wilson Cadamuro Júnior
*-----
SUBROUTINE DIM_EXATO (TOLERANCIA_Asi,N_SECAO,Z_SECAO,VETOR_N_SOLIC
+ ,VETOR_N_SOLIC_CONTRAV,VETOR_M1x_SOLIC,VETOR_M1y_SOLIC,
+ TIPO_PILAR_X,TIPO_PILAR_Y,K_MOLA_X,K_MOLA_Y,Nmax_POLIG_CONCR,
+ VETOR_N_POLIG_CONCR,Nmax_BARRAS_ACO,VETOR_N_BARRAS_ACO,
+ MATRIZ_X_POLIG_CONCR,MATRIZ_Y_POLIG_CONCR,MATRIZ_X_BARRAS_ACO,
+ MATRIZ_Y_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,fc,
+ FI_FLUENCIA,fy,Es,CLASSE_ACO,VETOR_Wx,VETOR_Wy,VETOR_M2x_SOLIC,
+ VETOR_M2y_SOLIC,Asi)
CHARACTER*1 CLASSE_ACO
CHARACTER*3 ROMPEU
CHARACTER*13 TIPO_RUPTURA
INTEGER SECAO_REF,N_SECAO,TIPO_PILAR_X,TIPO_PILAR_Y,
+ Nmax_POLIG_CONCR,VETOR_N_POLIG_CONCR(N_SECAO+1),Nmax_BARRAS_ACO,
+ VETOR_N_BARRAS_ACO(N_SECAO+1),N_m,VETOR_m(N_m+1),SECAO_RUPTURA,
+ N_POLIG_CONCR,N_ITERACOES_MAX,I,K,SECAO,BARRA,N_CALL,ya,yb,yx
INTEGER*2 HOR,MIN,SEG,DSEG
REAL*8 Z_SECAO(N_SECAO+1),VETOR_N_SOLIC(N_SECAO+1),
+ VETOR_N_SOLIC_CONTRAV(N_SECAO+1),VETOR_M1x_SOLIC(N_SECAO+1),
+ VETOR_M1y_SOLIC(N_SECAO+1),K_MOLA_X,K_MOLA_Y,
+ MATRIZ_X_POLIG_CONCR(N_SECAO+1,Nmax_POLIG_CONCR+1),
+ MATRIZ_Y_POLIG_CONCR(N_SECAO+1,Nmax_POLIG_CONCR+1),
+ MATRIZ_X_BARRAS_ACO(N_SECAO+1,Nmax_BARRAS_ACO+1),
+ MATRIZ_Y_BARRAS_ACO(N_SECAO+1,Nmax_BARRAS_ACO+1),
+ MATRIZ_A_BARRAS_ACO[ALLOCATABLE](:,:),
+ X_POLIG_CONCR[ALLOCATABLE](:),Y_POLIG_CONCR[ALLOCATABLE](:),
+ MATRIZ_C(N_m+1,30),MATRIZ_e_LIMITES(N_m+1,2),fc,FI_FLUENCIA,fy,
+ Es,VETOR_Wx(N_SECAO+1),VETOR_Wy(N_SECAO+1),
+ VETOR_M2x_SOLIC(N_SECAO+1),VETOR_M2y_SOLIC(N_SECAO+1),TIME1,
+ TIME2,DELTA_TIME,TOLERANCIA_Asi,Asi,a,b,x,w,M,Ac_REF
ALLOCATE (X_POLIG_CONCR(Nmax_POLIG_CONCR+1),
+ Y_POLIG_CONCR(Nmax_POLIG_CONCR+1),
+ MATRIZ_A_BARRAS_ACO(N_SECAO+1,Nmax_BARRAS_ACO+1))
* Obs.: a dimensão dos vetores contém o "+1" para não ocorrer erro no caso
* de se entrar com nmax_polig_concr ou nmax_barras_aco igual a zero!
CALL GETTIM(HOR,MIN,SEG,DSEG) !comandos para cálc. do tempo
TIME1=HOR*60*60+MIN*60+SEG+DSEG/100. ! de execução da subrotina
N_ITERACOES_MAX=10000 !número máximo de iterações
N_CALL=0

```

```

SECAO=1
SECAO_REF=SECAO                                !obtenção de uma seção de refer.
DO WHILE (VETOR_N_BARRAS_ACO(SECAO).EQ.0) ! a seção de referência deve ter
  SECAO=SECAO+1                                ! barras de aço; é em relação a
  SECAO_REF=SECAO                              ! esta seção que se calculará
  IF (SECAO.GT.N_SECAO) THEN                   ! w (taxa mecânica de armadura)
    CALL BEEPQQ(1500,1000)
    CALL BEEPQQ(1500,1000)                    !teste de consistência
    WRITE(*,*) 'ERRO (SUBROTINA DIM_EXATO): ',
+   'PILAR SEM BARRAS DE ACO'
    STOP
  END IF
END DO
I=1
N_POLIG_CONCR=VETOR_N_POLIG_CONCR(SECAO_REF)  !preparando as variáv.
DO WHILE (I.LE.N_POLIG_CONCR)                 ! para cálc. Ac_REF
  X_POLIG_CONCR(I)=MATRIZ_X_POLIG_CONCR(SECAO_REF,I)
  Y_POLIG_CONCR(I)=MATRIZ_Y_POLIG_CONCR(SECAO_REF,I)
  I=I+1
END DO                                         !cálc. da área da seção de referência (Ac_REF)
CALL GEOMETRIA (N_POLIG_CONCR,X_POLIG_CONCR,Y_POLIG_CONCR,Ac_REF,
+ a,b)
a=0                                           !tentativa inicial: w=0
w=a
Asi=w*Ac_REF*fc/fy/VETOR_N_BARRAS_ACO(SECAO_REF) !cálc. de Asi
SECAO=1
DO WHILE (SECAO.LE.N_SECAO)                  !atribui Asi a todas as barras de aço
  BARRA=1
  DO WHILE (BARRA.LE.VETOR_N_BARRAS_ACO(SECAO))
    MATRIZ_A_BARRAS_ACO(SECAO,BARRA)=Asi
    BARRA=BARRA+1
  END DO
  SECAO=SECAO+1
END DO
N_CALL=N_CALL+1                              !verifica se o pilar rompe ou não
CALL VERIFIC_EXATO (N_SECAO,Z_SECAO,VETOR_N_SOLIC,
+ VETOR_N_SOLIC_CONTRAV,VETOR_M1x_SOLIC,VETOR_M1y_SOLIC,
+ TIPO_PILAR_X,TIPO_PILAR_Y,K_MOLA_X,K_MOLA_Y,Nmax_POLIG_CONCR,
+ VETOR_N_POLIG_CONCR,Nmax_BARRAS_ACO,VETOR_N_BARRAS_ACO,
+ MATRIZ_X_POLIG_CONCR,MATRIZ_Y_POLIG_CONCR,MATRIZ_X_BARRAS_ACO,
+ MATRIZ_Y_BARRAS_ACO,MATRIZ_A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,
+ MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,VETOR_Wx,
+ VETOR_Wy,VETOR_M2x_SOLIC,VETOR_M2y_SOLIC,ROMPEU,TIPO_RUPTURA,
+ SECAO_RUPTURA)
IF (ROMPEU.EQ.'NAO') THEN
  ya=+1
  ELSE
  ya=-1
END IF
IF (ya.EQ.+1) GOTO 2                          !se a seção não rompeu com Asi=0, fim do processo
b=1
w=b                                           !segunda tentativa: w=1
Asi=w*Ac_REF*fc/fy/VETOR_N_BARRAS_ACO(SECAO_REF) !cálc. de Asi
SECAO=1
DO WHILE (SECAO.LE.N_SECAO)                  !atribui Asi a todas as barras de aço
  BARRA=1
  DO WHILE (BARRA.LE.VETOR_N_BARRAS_ACO(SECAO))
    MATRIZ_A_BARRAS_ACO(SECAO,BARRA)=Asi
    BARRA=BARRA+1
  END DO
  SECAO=SECAO+1
END DO
N_CALL=N_CALL+1                              !verifica se o pilar rompe ou não
CALL VERIFIC_EXATO (N_SECAO,Z_SECAO,VETOR_N_SOLIC,
+ VETOR_N_SOLIC_CONTRAV,VETOR_M1x_SOLIC,VETOR_M1y_SOLIC,
+ TIPO_PILAR_X,TIPO_PILAR_Y,K_MOLA_X,K_MOLA_Y,Nmax_POLIG_CONCR,
+ VETOR_N_POLIG_CONCR,Nmax_BARRAS_ACO,VETOR_N_BARRAS_ACO,
+ MATRIZ_X_POLIG_CONCR,MATRIZ_Y_POLIG_CONCR,MATRIZ_X_BARRAS_ACO,
+ MATRIZ_Y_BARRAS_ACO,MATRIZ_A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,
+ MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,VETOR_Wx,
+ VETOR_Wy,VETOR_M2x_SOLIC,VETOR_M2y_SOLIC,ROMPEU,TIPO_RUPTURA,
+ SECAO_RUPTURA)
IF (ROMPEU.EQ.'NAO') THEN
  yb=+1

```

```

ELSE
  yb=-1
END IF
K=0
DO WHILE (ya*yb.GT.0)                                !se com w=1 o pilar rompe...
  K=K+1                                              ! Método dos Intervalos
  a=b                                               ! Encaixantes até que o
  ya=yb                                             ! pilar não rompa
  b=b+1                                             ! incrementa w em 1
  w=b
  Asi=w*Ac_REF*fc/fy/VETOR_N_BARRAS_ACO(SECAO_REF) !cálc. de Asi
  SECAO=1
  DO WHILE (SECAO.LE.N_SECAO)                       !atribui Asi a todas as barras de aço
    BARRA=1
    DO WHILE (BARRA.LE.VETOR_N_BARRAS_ACO(SECAO))
      MATRIZ_A_BARRAS_ACO(SECAO,BARRA)=Asi
      BARRA=BARRA+1
    END DO
    SECAO=SECAO+1
  END DO
  N_CALL=N_CALL+1                                   !verifica se o pilar rompe ou não
  CALL VERIFIC_EXATO (N_SECAO,Z_SECAO,VETOR_N_SOLIC,
+   VETOR_N_SOLIC_CONTRAV,VETOR_M1x_SOLIC,VETOR_M1y_SOLIC,
+   TIPO_PILAR_X,TIPO_PILAR_Y,K_MOLA_X,K_MOLA_Y,Nmax_POLIG_CONCR,
+   VETOR_N_POLIG_CONCR,Nmax_BARRAS_ACO,VETOR_N_BARRAS_ACO,
+   MATRIZ_X_POLIG_CONCR,MATRIZ_Y_POLIG_CONCR,MATRIZ_X_BARRAS_ACO,
+   MATRIZ_Y_BARRAS_ACO,MATRIZ_A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,
+   MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,VETOR_Wx,
+   VETOR_Wy,VETOR_M2x_SOLIC,VETOR_M2y_SOLIC,ROMPEU,TIPO_RUPTURA,
+   SECAO_RUPTURA)
  IF (ROMPEU.EQ.'NAO') THEN
    yb=+1
  ELSE
    yb=-1
  END IF
  IF (K.GT.N_ITERACOES_MAX) THEN                     !teste de consistência
    CALL BEEPQQ(1500,1000)
    CALL BEEPQQ(1500,1000)
    WRITE(*,*) 'ERRO (SUBROTINA DIM_EXATO): ',
+   'ULTRAPASSADO O NUMERO MAXIMO DE ITERACOES'
    STOP
  END IF
END DO
K=1
M=ya                                               !início do loop do Método da Bissecção
1 x=(a+b)/2                                       !cálc. do novo w
w=x
Asi=w*Ac_REF*fc/fy/VETOR_N_BARRAS_ACO(SECAO_REF) !cálc. de Asi
SECAO=1
DO WHILE (SECAO.LE.N_SECAO)                       !atribui Asi a todas as barras de aço
  BARRA=1
  DO WHILE (BARRA.LE.VETOR_N_BARRAS_ACO(SECAO))
    MATRIZ_A_BARRAS_ACO(SECAO,BARRA)=Asi
    BARRA=BARRA+1
  END DO
  SECAO=SECAO+1
END DO
N_CALL=N_CALL+1                                   !verifica se o pilar rompe ou não
CALL VERIFIC_EXATO (N_SECAO,Z_SECAO,VETOR_N_SOLIC,
+   VETOR_N_SOLIC_CONTRAV,VETOR_M1x_SOLIC,VETOR_M1y_SOLIC,
+   TIPO_PILAR_X,TIPO_PILAR_Y,K_MOLA_X,K_MOLA_Y,Nmax_POLIG_CONCR,
+   VETOR_N_POLIG_CONCR,Nmax_BARRAS_ACO,VETOR_N_BARRAS_ACO,
+   MATRIZ_X_POLIG_CONCR,MATRIZ_Y_POLIG_CONCR,MATRIZ_X_BARRAS_ACO,
+   MATRIZ_Y_BARRAS_ACO,MATRIZ_A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,
+   MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,VETOR_Wx,
+   VETOR_Wy,VETOR_M2x_SOLIC,VETOR_M2y_SOLIC,ROMPEU,TIPO_RUPTURA,
+   SECAO_RUPTURA)
  IF (ROMPEU.EQ.'NAO') THEN
    yx=+1
  ELSE
    yx=-1
  END IF
  IF (M*yx.GT.0) THEN
    a=x

```



```

ya=yx
ELSE
b=x
yb=yx
END IF
!se o intervalo atende à tolerância...
IF (ABS(b-a).LE.TOLERANCIA_Asi*VETOR_N_BARRAS_ACO(SECAO_REF)*fy
+ /Ac_REF/fc) THEN
IF (x.NE.b) THEN !se o último ponto calculado for o w menor, recalcular
x=b ! com o w maior (a favor da segurança)
w=x
Asi=w*Ac_REF*fc/fy/VETOR_N_BARRAS_ACO(SECAO_REF) !cálc. de Asi
SECAO=1
DO WHILE (SECAO.LE.N_SECAO) !atribui Asi a todas as barras de aço
BARRA=1
DO WHILE (BARRA.LE.VETOR_N_BARRAS_ACO(SECAO))
MATRIZ_A_BARRAS_ACO(SECAO,BARRA)=Asi
BARRA=BARRA+1
END DO
SECAO=SECAO+1
END DO
N_CALL=N_CALL+1 !verifica se o pilar rompe ou não
CALL VERIFIC_EXATO (N_SECAO,Z_SECAO,VETOR_N_SOLIC,
+ VETOR_N_SOLIC_CONTRAV,VETOR_Mlx_SOLIC,VETOR_Mly_SOLIC,
+ TIPO_PILAR_X,TIPO_PILAR_Y,K_MOLA_X,K_MOLA_Y,Nmax_POLIG_CONCR
+ ,VETOR_N_POLIG_CONCR,Nmax_BARRAS_ACO,VETOR_N_BARRAS_ACO,
+ MATRIZ_X_POLIG_CONCR,MATRIZ_Y_POLIG_CONCR,
+ MATRIZ_X_BARRAS_ACO,MATRIZ_Y_BARRAS_ACO,MATRIZ_A_BARRAS_ACO,
+ N_m,VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,
+ CLASSE_ACO,VETOR_Wx,VETOR_Wy,VETOR_M2x_SOLIC,VETOR_M2y_SOLIC
+ ,ROMPEU,TIPO_RUPTURA,SECAO_RUPTURA)
IF (ROMPEU.EQ.'NAO') THEN
yx=+1
ELSE
yx=-1
END IF
END IF
GOTO 2 !fim do processo
END IF
K=K+1
IF (K.GT.N_ITERACOES_MAX) THEN !teste de consistência
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) 'ERRO (SUBROTINA DIM_EXATO): ',
+ 'ULTRAPASSADO O NUMERO MAXIMO DE ITERACOES'
STOP
END IF
GOTO 1
*cálculos e testes finais da subrotina:
2 CALL GETTIM(HOR,MIN,SEG,DSEG) !comandos para cálc. do tempo
TIME2=HOR*60*60+MIN*60+SEG+DSEG/100. ! de execução da subrotina
IF (TIME2.LT.TIME1) TIME2=TIME2+24*60*60
DELTA_TIME=(TIME2-TIME1)/60.
WRITE(*,3) K,N_CALL,DELTA_TIME
3 FORMAT(' DIM_EXATO: ',I4,' iteracoes ',I4,
+ ' chamadas ',F8.2,' minutos') !mensagem informativa
IF (ROMPEU.EQ.'SIM') THEN !teste de consistência
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) 'ERRO (SUBROTINA DIM_EXATO): ',
+ 'ARMADURA CALCULADA EH INSUFICIENTE'
STOP
END IF
DEALLOCATE (X_POLIG_CONCR,Y_POLIG_CONCR,MATRIZ_A_BARRAS_ACO)
END
=====fim da subrotina DIM_EXATO=====

```

```

=====DIM_PPADRAO=====
*Subrotina que, dado um pilar totalmente definido (menos a área das barras da
* armadura) e seus esforços solicitantes, calcula a menor área de aço capaz de
* tornar o pilar estável. O cálculo é feito por tentativas: arbitra-se taxas de
* armadura e utiliza-se a subrotina verific_ppadrao para verificar se tal
* pilar rompe ou não. Para otimizar este processo, esta subrotina utiliza um

```

* método numérico iterativo de cálculo de zeros de funções: o Método da Bisseção (ver Ruggiero, M.A.G. e Lopes, V.L.R. - "Cálculo Numérico - aspectos teóricos e computacionais"). Este método foi escolhido por ser eficiente e por ter convergência garantida para o tipo de função em questão rompu=f(Asi), ao contrário de outros métodos mais eficientes, como o de Newton-Raphson ou o da Secante. Para o entendimento da subrotina, recomenda-se consultar a bibliografia indicada acima. O pilar deve estar referenciado num sistema de coordenadas Oxyz dextrógiro, com seções transversais no plano xy e eixo do pilar ao longo do eixo z; a subrotina verific_ppadrao utiliza o Método do Equilíbrio com o Processo do Pilar Padrão (veja as características e restrições de tal método em FUSCO, P.B. (1981) "Estruturas de Concreto - Solicitações Normais").

*Unidades de entrada: Sistema de unidades qualquer, desde que consistente;

*TOLERANCIA_Asi: Tolerância com que se calculará Asi (veja a definição de Asi abaixo); o Asi calculado pode conter um erro a favor da segurança de, no máximo, tolerancia_asi; a unidade de entrada deve ter dimensão L2 (área);

*N_SOLIC, Mlx_SOLIC, Mly_SOLIC: Esforços de primeira ordem que solicitam a seção de referência o pilar (valores de cálculo):

* N_SOLIC >0 -> força normal de compressão;

* Mlx_SOLIC >0 -> momento definido por uma excentricidade positiva no eixo x;

* Mly_SOLIC >0 -> momento definido por uma excentricidade positiva no eixo y;

* não há restrições quanto à esses valores, que podem ser positivos, negativos ou zero;

*N_SOLIC_CONTRAV: No caso de o pilar em análise ser uma estrutura de contraventamento (núcleo, pilar parede, etc.), N_SOLIC_CONTRAV é a força normal total que age na estrutura contraventada (valor de cálculo); convenção de sinal: positivo se for de compressão;

*L_fl: Comprimento de flambagem do pilar;

*N_POLIG_CONCR e N_BARRAS_ACO: Número de nós da poligonal de concreto (incluindo o nó de chegada, que é o mesmo nó de saída) e número de barras de aço da seção;

*X_POLIG_CONCR, Y_POLIG_CONCR e X_BARRAS_ACO, Y_BARRAS_ACO: Vetores das coordenadas (sistema Oxy, ou seja, não rotacionado) dos nós da poligonal de concreto (a poligonal deve ser numerada no sentido anti-horário, com vazios no sentido horário; com o nó de chegada, que é o mesmo nó de saída) e das coordenadas das barras de aço;

*N_m: Número de curvas (polinômios de grau qualquer) que compoem o diagrama tensão-deformação adotado para o concreto; cada curva é definida por um somatório do tipo: $\sigma_c = f_c(C_0.e^0 + C_1.e^1 + C_2.e^2 + \dots + C_m.e^m)$; para o diagrama parábola-retângulo da NB1/78, temos $N_m=2$ (curva parabólica e curva reta constante);

*VETOR_m: Vetor que contém os graus dos N_m polinômios; para o parábola-retângulo da NB1/78, temos $VETOR_m=(2,0)$ ($m=2$ -> parábola; $m=0$ -> reta constante);

*MATRIZ_C: Matriz que contém os coeficientes dos N_m polinômios: linha i contém os C_k ($k=1$ à $m+1$) coeficientes do polinômio i; para o parábola-retângulo da NB1/78, temos: $MATRIZ_C = \begin{matrix} !0 & 1 & -0.25! & & -> \text{coef. da parábola} \\ & !1 & 0 & 0 & ! & -> \text{coef. da reta constante;} \end{matrix}$

*MATRIZ_e_LIMITES: Matriz que contém as deformações (em por mil) iniciais e finais onde age cada um dos N_m polinômios (deformações estas sem o efeito da fluência): linha i contém as deformações iniciais e finais onde age o polinômio i; OBS: colocar um número grande para a deformação final do último polinômio que compõe o diagrama tensão-deformação adotado para o concreto, não finalizando o diagrama no ELU; para o parábola-retângulo da NB1/78, temos: $MATRIZ_e_LIMITES = \begin{matrix} !0 & & 2 & ! & -> \text{def. limites da parábola} \\ & & !2 & 1000! & -> \text{def. limites da reta constante;} \end{matrix}$

*fc: Tensão máxima de compressão à que o concreto pode estar sujeito, ou seja, é a tensão máxima do diagrama tensão-deformação adotado para o concreto (valor de cálculo). Não há consenso quanto ao valor a se utilizar. A NB1/78 indica $fc=0.85fcd$ (ELU), mas, para o estudo da estabilidade, alguns autores indicam $fc=0.85fck$.

*FI_FLUENCIA: Função de Fluência (Teoria Linear de Fluência);

*fy: Tensão de escoamento do aço (valor de cálculo);

*Es: Módulo de Elasticidade do aço (usualmente igual a 210000MPa);

*CLASSE_ACO: É a classe à que pertence o aço (A ou B);

*e2x, e2y: Saídas da subrotina: Excentricidades de segunda ordem na seção de referência pilar;

*M2x_SOLIC, M2y_SOLIC: Saídas da subrotina: Esforços de segunda ordem que solicitam a seção de referência do pilar (valores de cálculo):

* M2x_SOLIC >0 -> momento definido por uma excentricidade positiva no eixo x;

* M2y_SOLIC >0 -> momento definido por uma excentricidade positiva no eixo y;

*Asi: Área de uma barra de aço (a seção tem um total de n_barras_aco);

* considera-se que todas as barras de aço da seção tem a mesma área.

* SISTEMA FLEXOR - DIMENSIONAMENTO E VERIFICAÇÃO DE PILARES ESBELTOS DE

```

*           CONCRETO ARMADO COM SEÇÃO QUALQUER SOLICITADOS POR FLEXÃO
*           COMPOSTA OBLÍQUA
*           DIAG. TENSÃOxDEFORM.DO CONCRETO: POLINÔMIOS DE GRAU QUALQUER
*           CONSIDERAÇÃO DA FLUÊNCIA: TEORIA LINEAR DE FLUÊNCIA
*           AÇOS BRASILEIROS DO TIPO 'A' OU 'B' COM QUALQUER fy OU Es
*           Copyright(c) 1997, Ismael Wilson Cadamuro Júnior
* -----

```

```

SUBROUTINE DIM_PPADRAO (TOLERANCIA_Asi,N_SOLIC,N_SOLIC_CONTRAV,
+ M1x_SOLIC,M1y_SOLIC,L_fl,N_POLIG_CONCR,N_BARRAS_ACO,
+ X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,Y_BARRAS_ACO,N_m,
+ VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,
+ CLASSE_ACO,e2x,e2y,M2x_SOLIC,M2y_SOLIC,Asi)
CHARACTER*1 CLASSE_ACO
CHARACTER*3 ROMPEU
CHARACTER*13 TIPO_RUPTURA
INTEGER N_POLIG_CONCR,N_BARRAS_ACO,N_m,VETOR_m(N_m+1),
+ N_ITERACOES_MAX,K,I,ya,yb,yx,N_CALL
INTEGER*2 HOR,MIN,SEG,DSEG
REAL*8 N_SOLIC,N_SOLIC_CONTRAV,M1x_SOLIC,M1y_SOLIC,M2x_SOLIC,
+ M2y_SOLIC,L_fl,X_POLIG_CONCR(N_POLIG_CONCR+1),
+ Y_POLIG_CONCR(N_POLIG_CONCR+1),X_BARRAS_ACO(N_BARRAS_ACO+1),
+ Y_BARRAS_ACO(N_BARRAS_ACO+1),MATRIZ_C(N_m+1,30),
+ MATRIZ_e_LIMITES(N_m+1,2),fc,FI_FLUENCIA,fy,Es,Asi,TIME1,TIME2,
+ DELTA_TIME,TOLERANCIA_Asi,a,b,x,w,M,A_BARRAS_ACO[ALLOCATABLE](:),
+ ,Ac,e2x,e2y
*   Obs.: a dimensão dos vetores contém o "+1" para não ocorrer erro no caso
*   de se entrar com n_polig_concr ou n_barras_aco igual a zero!
CALL GETTIM(HOR,MIN,SEG,DSEG)           !comandos para cálc. do tempo
TIME1=HOR*60*60+MIN*60+SEG+DSEG/100.    !   de execução da subrotina
ALLOCATE (A_BARRAS_ACO(N_BARRAS_ACO+1))
N_ITERACOES_MAX=10000                   !número máximo de iterações
N_CALL=0                                 !cálc. da área da seção (Ac)
CALL GEOMETRIA (N_POLIG_CONCR,X_POLIG_CONCR,Y_POLIG_CONCR,Ac,a,b)
a=0
w=a                                       !tentativa inicial: w=0
Asi=w*Ac*fc/fy/N_BARRAS_ACO             !cálc. de Asi
I=1
DO WHILE (I.LE.N_BARRAS_ACO)             !atribui Asi a todas as barras de aço
  A_BARRAS_ACO(I)=Asi
  I=I+1
END DO
N_CALL=N_CALL+1                           !verifica se o pilar rompeu ou não
CALL VERIFIC_PPADRAO (N_SOLIC,N_SOLIC_CONTRAV,M1x_SOLIC,M1y_SOLIC,
+ L_fl,N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,
+ X_BARRAS_ACO,Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,
+ MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,e2x,e2y,
+ M2x_SOLIC,M2y_SOLIC,ROMPEU,TIPO_RUPTURA)
IF (ROMPEU.EQ.'NAO') THEN
  ya=+1
  ELSE
  ya=-1
END IF
IF (ya.EQ.+1) GOTO 2                       !se a seção não rompeu com Asi=0, fim do processo
b=1
w=b                                       !segunda tentativa: w=1
Asi=w*Ac*fc/fy/N_BARRAS_ACO             !cálc. de Asi
I=1
DO WHILE (I.LE.N_BARRAS_ACO)             !atribui Asi a todas as barras de aço
  A_BARRAS_ACO(I)=Asi
  I=I+1
END DO
N_CALL=N_CALL+1                           !verifica se o pilar rompeu ou não
CALL VERIFIC_PPADRAO (N_SOLIC,N_SOLIC_CONTRAV,M1x_SOLIC,M1y_SOLIC,
+ L_fl,N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,
+ X_BARRAS_ACO,Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,
+ MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,e2x,e2y,
+ M2x_SOLIC,M2y_SOLIC,ROMPEU,TIPO_RUPTURA)
IF (ROMPEU.EQ.'NAO') THEN
  yb=+1
  ELSE
  yb=-1
END IF
K=0
DO WHILE (ya*yb.GT.0)                       !se com w=1 o pilar rompe...

```

```

K=K+1 ! Método dos Intervalos
a=b ! Encaixantes até que o
ya=yb ! pilar não rompa
b=b+1 ! incrementa w em 1
w=b
Asi=w*Ac*fc/fy/N_BARRAS_ACO !cálc. de Asi
I=1
DO WHILE (I.LE.N_BARRAS_ACO) !atribui Asi a todas as barras de aço
A_BARRAS_ACO(I)=Asi
I=I+1
END DO
N_CALL=N_CALL+1 !verifica se o pilar rompe ou não
CALL VERIFIC_PPADRAO (N_SOLIC,N_SOLIC_CONTRAV,M1x_SOLIC,
+ M1y_SOLIC,L_fl,N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,
+ Y_POLIG_CONCR,X_BARRAS_ACO,Y_BARRAS_ACO,A_BARRAS_ACO,N_m,
+ VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,
+ CLASSE_ACO,e2x,e2y,M2x_SOLIC,M2y_SOLIC,ROMPEU,TIPO_RUPTURA)
IF (ROMPEU.EQ.'NAO') THEN
yb=+1
ELSE
yb=-1
END IF
IF (K.GT.N_ITERACOES_MAX) THEN !teste de consistência
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) 'ERRO (SUBROTINA DIM_PPADRAO): ',
+ 'ULTRAPASSADO O NUMERO MAXIMO DE ITERACOES'
STOP
END IF
END DO
K=1
M=ya !início do loop do Método da Bisseção
1 x=(a+b)/2 !cálc. do novo w
w=x
Asi=w*Ac*fc/fy/N_BARRAS_ACO !cálc. de Asi
I=1
DO WHILE (I.LE.N_BARRAS_ACO) !atribui Asi a todas as barras de aço
A_BARRAS_ACO(I)=Asi
I=I+1
END DO
N_CALL=N_CALL+1 !verifica se o pilar rompe ou não
CALL VERIFIC_PPADRAO (N_SOLIC,N_SOLIC_CONTRAV,M1x_SOLIC,M1y_SOLIC,
+ L_fl,N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,
+ X_BARRAS_ACO,Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,
+ MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,e2x,e2y,
+ M2x_SOLIC,M2y_SOLIC,ROMPEU,TIPO_RUPTURA)
IF (ROMPEU.EQ.'NAO') THEN
yx=+1
ELSE
yx=-1
END IF
IF (M*yx.GT.0) THEN
a=x
ya=yx
ELSE
b=x
yb=yx
END IF
!se o intervalo atende à tolerância...
IF (ABS(b-a).LE.TOLERANCIA_Asi*N_BARRAS_ACO*fy/Ac/fc) THEN
IF (x.NE.b) THEN !se o último ponto calculado for o w menor, recalcular
x=b ! com o w maior (a favor da segurança)
w=x
Asi=w*Ac*fc/fy/N_BARRAS_ACO !cálc. de Asi
I=1
DO WHILE (I.LE.N_BARRAS_ACO) !atribui Asi a todas as barras de aço
A_BARRAS_ACO(I)=Asi
I=I+1
END DO
N_CALL=N_CALL+1 !verifica se o pilar rompe ou não
CALL VERIFIC_PPADRAO (N_SOLIC,N_SOLIC_CONTRAV,M1x_SOLIC,
+ M1y_SOLIC,L_fl,N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,
+ Y_POLIG_CONCR,X_BARRAS_ACO,Y_BARRAS_ACO,A_BARRAS_ACO,N_m,
+ VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,
+ CLASSE_ACO,e2x,e2y,M2x_SOLIC,M2y_SOLIC,ROMPEU,TIPO_RUPTURA)

```

```

        IF (ROMPEU.EQ.'NAO') THEN
            yx=+1
        ELSE
            yx=-1
        END IF
    END IF
    GOTO 2                                !fim do processo
END IF
K=K+1
IF (K.GT.N_ITERACOES_MAX) THEN          !teste de consistência
    CALL BEEPQQ(1500,1000)
    CALL BEEPQQ(1500,1000)
    WRITE(*,*) 'ERRO (SUBROTINA DIM_PPADRAO): ',
+ 'ULTRAPASSADO O NUMERO MAXIMO DE ITERACOES'
    STOP
END IF
GOTO 1
*cálculos e testes finais da subrotina:
2 CALL GETTIM(HOR,MIN,SEG,DSEG)          !comandos para cálc. do tempo
   TIME2=HOR*60*60+MIN*60+SEG+DSEG/100. ! de execução da subrotina
   IF (TIME2.LT.TIME1) TIME2=TIME2+24*60*60
   DELTA_TIME=(TIME2-TIME1)/60.
   WRITE(*,3) K,N_CALL,DELTA_TIME
3 FORMAT(' DIM_PPADRAO: ',I4,' iteracoes ',I4
+ ' chamadas ',F8.2,' minutos')        !mensagem informativa
   IF (ROMPEU.EQ.'SIM') THEN           !teste de consistência
       CALL BEEPQQ(1500,1000)
       CALL BEEPQQ(1500,1000)
       WRITE(*,*) 'ERRO (SUBROTINA DIM_PPADRAO): ',
+ 'ARMADURA CALCULADA EH INSUFICIENTE'
       STOP
   END IF
   DEALLOCATE (A_BARRAS_ACO)
END
=====fim da subrotina DIM_PPADRAO=====

```

```

=====DIM_ELU=====
*Subrotina que, dada uma seção transversal totalmente definida (menos a área das
*barras da armadura) e seus esforços solicitantes, calcula a menor área de aço
* capaz de fazer com que a seção não rompa. O cálculo é feito por tentativas:
* arbitra-se taxas de armadura e utiliza-se a subrotina verific_elu para
* verificar se tal seção rompe ou não. Para otimizar este processo, esta
* subrotina utiliza um método numérico iterativo de cálculo de zeros de
* funções: o Método da Bisseção (ver Ruggiero, M.A.G. e Lopes, V.L.R. -
* "Cálculo Numérico - aspectos teóricos e computacionais"). Este método foi
* escolhido por ser eficiente e por ter convergência garantida para o tipo de
* função em questão rompeu=f(Asi), ao contrário de outros métodos mais
* eficientes, como o de Newton-Raphson ou o da Secante. Para o entendimento da
* subrotina, recomenda-se consultar a bibliografia indicada acima.
*Unidades de entrada: Sistema de unidades qualquer, desde que consistente;
*TOLERANCIA_Asi: Tolerância com que se calculará Asi (veja a definição de Asi
* abaixo); o Asi calculado pode conter um erro a favor da segurança de, no
* máximo, tolerancia_asi; a unidade de entrada deve ter dimensão L2 (área);
*N_SOLIC, Mx_SOLIC, My_SOLIC: Esforços que solicitam a seção (valores de
* cálculo):
* N_SOLIC >0 -> força normal de compressão;
* Mx_SOLIC >0 -> momento definido por uma excentricidade positiva no eixo x;
* My_SOLIC >0 -> momento definido por uma excentricidade positiva no eixo y;
* não há restrições quanto à esses valores, que podem ser positivos, negativos
* ou zero;
*N_POLIG_CONCR e N_BARRAS_ACO: Número de nós da poligonal de concreto (incluindo
* o nó de chegada, que é o mesmo nó de saída) e número de barras de aço da
* seção;
*X_POLIG_CONCR, Y_POLIG_CONCR e X_BARRAS_ACO, Y_BARRAS_ACO: Vetores das
* coordenadas (sistema Oxy, ou seja, não rotacionado) dos nós da poligonal de
* concreto (a poligonal deve ser numerada no sentido anti-horário, com vazios
* no sentido horário; com o nó de chegada, que é o mesmo nó de saída) e das
* coordenadas das barras de aço;
*A_BARRAS_ACO: Vetor das áreas das barras de aço;
*N_m: Número de curvas (polinômios de grau qualquer) que compoem o diagrama
* tensão-deformação adotado para o concreto; cada curva é definida por um
* somatório do tipo: sigma_c=fc(C0.e^0 + C1.e^1 + C2.e^2 +...+Cm.e^m); para o

```

```

* diagrama parábola-retângulo da NBl/78, temos N_m=2 (curva parabólica e curva
* reta constante);
*VETOR_m: Vetor que contém os graus dos N_m polinômios; para o parábola-retângulo
* da NBl/78, temos VETOR_m=(2,0) (m=2 -> parábola; m=0 -> reta constante);
*MATRIZ_C: Matriz que contém os coeficientes dos N_m polinômios: linha i contém
* os Ck (k=1 à m+1) coeficientes do polinômio i; para o parábola-retângulo da
* NBl/78, temos: MATRIZ_C = !0 1 -0.25! -> coef. da parábola
* !1 0 0 ! -> coef. da reta constante;
*MATRIZ_e_LIMITES: Matriz que contém as deformações (em por mil) iniciais e
* finais onde age cada um dos N_m polinômios (deformações estas sem o efeito da
* fluência): linha i contém as deformações iniciais e finais onde age o
* polinômio i; OBS: colocar um número grande para a deformação final do último
* polinômio que compõe o diagrama tensão-deformação adotado para o concreto,
* não finalizando o diagrama no ELU; para o parábola-retângulo da NBl/78, temos:
* MATRIZ_e_LIMITES = !0 2 ! -> def. limites da parábola
* !2 1000! -> def. limites da reta constante;
*fc: Tensão máxima de compressão à que o concreto pode estar sujeito, ou seja,
* é a tensão máxima do diagrama tensão-deformação adotado para o concreto.
* A NBl/78 indica fc=0.85fcd (ELU).
*FI_FLUENCIA: Função de Fluência (Teoria Linear de Fluência);
*fy: Tensão de escoamento do aço (valor de cálculo);
*Es: Módulo de Elasticidade do aço (usualmente igual a 210000MPa);
*CLASSE_ACO: É a classe à que pertence o aço (A ou B);
*ALFAL,2:Saídas da subrotina: Ângulo de rotação da linha neutra que possibilita
* beta_r ser igual ao beta_s dado; alfa é medido a partir do eixo x no
* sentido trigonométrico (anti-horário), e expresso em radianos; alfa pode
* variar de -pi à +pi radianos; o índice 1 ou 2 refere-se aos problemas com
* duas respostas (conforme explicado no início da subrotina verific_elu);
* caso o problema só admita uma resposta, alfa1=0;
*H_ALFAL,2:Saídas da subrotina: Altura máxima da poligonal de concreto
* perpendicular à direção da linha neutra (alfa+90 graus); o índice 1 ou 2
* refere-se aos problemas com duas respostas (conforme explicado no início da
* subrotina verific_elu); caso o problema só admita uma resposta, h_alfa1=0;
*TETA1,2 e e_CG1,2: Saídas da subrotina: Curvatura Majorada Adimensional e
* Deformação do CG que possibilitam que a força normal resistente seja igual à
* força normal solicitante, com a seção atingindo o ELU de ruptura;
* Teta=1000h_alfa/r (na direção perpendicular à da linha neutra, encurtando
* mais as fibras superiores à linha neutra); e_cg é expresso em por mil e
* convencionalmente positivo se for encurtamento; admite-se que o CG esteja
* localizado na origem do sistema de coordenadas: é em relação a este ponto que
* são calculados os esforços resistentes da seção; o índice 1 ou 2 refere-se
* aos problemas com duas respostas (conforme explicado no início da subrotina
* verific_elu); caso o problema só admita uma resposta, teta1=0 e e_CG1=0;
*Mrx1,2, Mry1,2: Saídas da subrotina: Momentos flettores últimos da seção:
* Mrx>0 --> momento definido por uma excentricidade positiva sobre o eixo x;
* Mry>0 --> momento definido por uma excentricidade positiva sobre o eixo y;
* o índice 1 ou 2 refere-se aos problemas com duas respostas (conforme
* explicado no início da subrotina verific_elu); o índice 2 indica o ponto de
* maior módulo do momento fletor resistente, e o 1 o de menor; caso o problema
* só admita uma resposta, Mrx1=0 e Mry1=0; a direção (beta_r) dos momentos 1 e
* 2 é a mesma direção do momento solicitante dado (beta_s);
*-----
* SISTEMA FLEXOR - DIMENSIONAMENTO E VERIFICAÇÃO DE PILARES ESBELTOS DE
* CONCRETO ARMADO COM SEÇÃO QUALQUER SOLICITADOS POR FLEXÃO
* COMPOSTA OBLÍQUA
* DIAG. TENSÃOxDEFORM.DO CONCRETO: POLINÔMIOS DE GRAU QUALQUER
* CONSIDERAÇÃO DA FLUÊNCIA: TEORIA LINEAR DE FLUÊNCIA
* AÇOS BRASILEIROS DO TIPO 'A' OU 'B' COM QUALQUER fy OU Es
* Copyright(c) 1997, Ismael Wilson Cadamuro Júnior
*-----
SUBROUTINE DIM_ELU (TOLERANCIA_Asi,N_SOLIC, Mx_SOLIC, My_SOLIC,
+ N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,
+ X_BARRAS_ACO,Y_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,
+ fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,TETA1,ALFAL,H_ALFAL,e_CG1,Mrx1,
+ Mry1,TETA2,ALFA2,H_ALFA2,e_CG2,Mrx2,Mry2,Asi)
CHARACTER*1 CLASSE_ACO
CHARACTER*3 EXISTE,ROMPEU
INTEGER N_POLIG_CONCR,N_BARRAS_ACO,N_m,VETOR_m(N_m+1),N_CALL,
+ N_ITERACOES_MAX,I,K,ya,yb,yx
INTEGER*2 HOR,MIN,SEG,DSEG
REAL*8 N_SOLIC,X_POLIG_CONCR(N_POLIG_CONCR+1),
+ Y_POLIG_CONCR(N_POLIG_CONCR+1),X_BARRAS_ACO(N_BARRAS_ACO+1),
+ Y_BARRAS_ACO(N_BARRAS_ACO+1),MATRIZ_C(N_m+1,30),
+ MATRIZ_e_LIMITES(N_m+1,2),fc,FI_FLUENCIA,fy,Es,TETA1,ALFAL,

```

```

+ H_ALFA1,e_CG1,Mrx1,Mry1,TETA2,ALFA2,H_ALFA2,e_CG2,Mrx2,Mry2,
+ TIME1,TIME2,DELTA_TIME,Mx_SOLIC,My_SOLIC,TOLERANCIA_Asi,Asi,a,b,
+ x,w,M,A_BARRAS_ACO[ALLOCATABLE](:),Ac
*
* Obs.: a dimensão dos vetores contém o "+1" para não ocorrer erro no caso
de se entrar com n_polig_concr ou n_barras_aco igual a zero!
CALL GETTIM(HOR,MIN,SEG,DSEG) !comandos para cálc. do tempo
TIME1=HOR*60*60+MIN*60+SEG+DSEG/100. ! de execução da subrotina
ALLOCATE (A_BARRAS_ACO(N_BARRAS_ACO+1))
N_ITERACOES_MAX=10000 !número máximo de iterações
N_CALL=0 !cálc. da área da seção (Ac)
CALL GEOMETRIA (N_POLIG_CONCR,X_POLIG_CONCR,Y_POLIG_CONCR,Ac,a,b)
a=0
w=a !tentativa inicial: w=0
Asi=w*Ac*fc/fy/N_BARRAS_ACO !cálc. de Asi
I=1
DO WHILE (I.LE.N_BARRAS_ACO) !atribui Asi a todas as barras de aço
A_BARRAS_ACO(I)=Asi
I=I+1
END DO
N_CALL=N_CALL+1 !verifica se o pilar rompe ou não
CALL VERIFIC_ELU (N_SOLIC, Mx_SOLIC, My_SOLIC,N_POLIG_CONCR,
+ N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,
+ Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,
+ fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,TETA1,ALFA1,H_ALFA1,e_CG1,Mrx1,
+ Mry1,TETA2,ALFA2,H_ALFA2,e_CG2,Mrx2,Mry2,EXISTE,ROMPEU)
IF (ROMPEU.EQ.'NAO') THEN
ya=+1
ELSE
ya=-1
END IF
IF (ya.EQ.+1) GOTO 2 !se a seção não rompeu com Asi=0, fim do processo
b=1
w=b !segunda tentativa: w=1
Asi=w*Ac*fc/fy/N_BARRAS_ACO !cálc. de Asi
I=1
DO WHILE (I.LE.N_BARRAS_ACO) !atribui Asi a todas as barras de aço
A_BARRAS_ACO(I)=Asi
I=I+1
END DO
N_CALL=N_CALL+1 !verifica se o pilar rompe ou não
CALL VERIFIC_ELU (N_SOLIC, Mx_SOLIC, My_SOLIC,N_POLIG_CONCR,
+ N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,
+ Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,
+ fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,TETA1,ALFA1,H_ALFA1,e_CG1,Mrx1,
+ Mry1,TETA2,ALFA2,H_ALFA2,e_CG2,Mrx2,Mry2,EXISTE,ROMPEU)
IF (ROMPEU.EQ.'NAO') THEN
yb=+1
ELSE
yb=-1
END IF
K=0
DO WHILE (ya*yb.GT.0) !se com w=1 o pilar rompe...
K=K+1 ! Método dos Intervalos
a=b ! Encaixantes até que o
ya=yb ! pilar não rompa
b=b+1 !incrementa w em 1
w=b
Asi=w*Ac*fc/fy/N_BARRAS_ACO !cálc. de Asi
I=1
DO WHILE (I.LE.N_BARRAS_ACO) !atribui Asi a todas as barras de aço
A_BARRAS_ACO(I)=Asi
I=I+1
END DO
N_CALL=N_CALL+1 !verifica se o pilar rompe ou não
CALL VERIFIC_ELU (N_SOLIC, Mx_SOLIC, My_SOLIC,N_POLIG_CONCR,
+ N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,
+ Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,
+ MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,TETA1,ALFA1,
+ H_ALFA1,e_CG1,Mrx1,Mry1,TETA2,ALFA2,H_ALFA2,e_CG2,Mrx2,Mry2,
+ EXISTE,ROMPEU)
IF (ROMPEU.EQ.'NAO') THEN
yb=+1
ELSE
yb=-1

```

```

        END IF
        IF (K.GT.N_ITERACOES_MAX) THEN                                !teste de consistência
            CALL BEEPQQ(1500,1000)
            CALL BEEPQQ(1500,1000)
            WRITE(*,*) 'ERRO (SUBROTINA DIM_ELU): ',
+ 'ULTRAPASSADO O NUMERO MAXIMO DE ITERACOES'
            STOP
        END IF
    END DO
    K=1
    M=ya                                                                !início do loop do Método da Bissecção
1  x=(a+b)/2                                                            !cálc. do novo w
    w=x
    Asi=w*Ac*fc/fy/N_BARRAS_ACO                                       !cálc. de Asi
    I=1
    DO WHILE (I.LE.N_BARRAS_ACO)                                       !atribui Asi a todas as barras de aço
        A_BARRAS_ACO(I)=Asi
        I=I+1
    END DO
    N_CALL=N_CALL+1                                                    !verifica se o pilar rompe ou não
    CALL VERIFIC_ELU (N_SOLIC, Mx_SOLIC, My_SOLIC,N_POLIG_CONCR,
+ N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,
+ Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,
+ fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,TETA1,ALFA1,H_ALFA1,e_CG1,Mrx1,
+ Mry1,TETA2,ALFA2,H_ALFA2,e_CG2,Mrx2,Mry2,EXISTE,ROMPEU)
    IF (ROMPEU.EQ.'NAO') THEN
        yx=+1
    ELSE
        yx=-1
    END IF
    IF (M*yx.GT.0) THEN
        a=x
        ya=yx
    ELSE
        b=x
        yb=yx
    END IF
    !se o intervalo atende à tolerância...
    IF (ABS(b-a).LE.TOLERANCIA_Asi*N_BARRAS_ACO*fy/Ac/fc) THEN
        IF (x.NE.b) THEN !se o último ponto calculado for o w menor, recalcular
            x=b          ! com o w maior (a favor da segurança)
            w=x
            Asi=w*Ac*fc/fy/N_BARRAS_ACO           !cálc. de Asi
            I=1
            DO WHILE (I.LE.N_BARRAS_ACO)         !atribui Asi a todas as barras de aço
                A_BARRAS_ACO(I)=Asi
                I=I+1
            END DO
            N_CALL=N_CALL+1                       !verifica se o pilar rompe ou não
            CALL VERIFIC_ELU (N_SOLIC, Mx_SOLIC, My_SOLIC,N_POLIG_CONCR,
+ N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,
+ Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,
+ MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,TETA1,ALFA1
+ ,H_ALFA1,e_CG1,Mrx1,Mry1,TETA2,ALFA2,H_ALFA2,e_CG2,Mrx2,Mry2
+ ,EXISTE,ROMPEU)
            IF (ROMPEU.EQ.'NAO') THEN
                yx=+1
            ELSE
                yx=-1
            END IF
        END IF
        GOTO 2                                                                !fim do processo
    END IF
    K=K+1
    IF (K.GT.N_ITERACOES_MAX) THEN                                !teste de consistência
        CALL BEEPQQ(1500,1000)
        CALL BEEPQQ(1500,1000)
        WRITE(*,*) 'ERRO (SUBROTINA DIM_ELU): ',
+ 'ULTRAPASSADO O NUMERO MAXIMO DE ITERACOES'
        STOP
    END IF
    GOTO 1

```

*cálculos e testes finais da subrotina:

```

2  CALL GETTIM(HOR,MIN,SEG,DSEG)                                !comandos para cálc. do tempo
    TIME2=HOR*60*60+MIN*60+SEG+DSEG/100.                       ! de execução da subrotina

```



```

        IF (TIME2.LT.TIME1) TIME2=TIME2+24*60*60
        DELTA_TIME=(TIME2-TIME1)/60.
        WRITE(*,3) K,N_CALL,DELTA_TIME
3   FORMAT(' DIM_ELU: ',I4,' iteracoes      ',I4
+   ' chamadas      ',F8.2,' minutos')           !mensagem informativa
        IF (ROMPEU.EQ.'SIM') THEN                 !teste de consistência
            CALL BEEPQQ(1500,1000)
            CALL BEEPQQ(1500,1000)
            WRITE(*,*) 'ERRO (SUBROTINA DIM_ELU): ',
+   ' ARMADURA CALCULADA EH INSUFICIENTE.'
            PAUSE
        END IF
        DEALLOCATE (A_BARRAS_ACO)
        END
=====fim da subrotina DIM_ELU=====

```

```

=====VERIFIC_EXATO=====
*Subrotina que, dado um pilar totalmente definido e seus esforços solicitantes,
* verifica se tal pilar rompe ou não, e calcula a posição deformada do pilar
* (caso este não rompa). Para isso, utiliza o Método do Equilíbrio com o
* Processo Exato (veja as características e restrições de tal método em
* FUSCO,P.B.(1981) "Estruturas de Concreto - Solicitações Normais"). O pilar
* deve estar referenciado num sistema de coordenadas Oxyz dextrógiro, com
* seções transversais no plano xy e eixo do pilar ao longo do eixo z.
* Unidades de entrada: Sistema de unidades qualquer, desde que consistente;
*N_SECAO: Número de seções transversais do pilar; o pilar deve ser discretizado
* com seções numeradas de 1 a N_SECAO; N_SECAO>=2;
*Z_SECAO: Vetor (N_SECAOx1) que contém as coordenadas z de cada seção
* transversal; a seção 1 deve ter z=0, e as demais z>0; duas seções podem ter o
* mesmo z (caso de pilar com variação brusca de seção ou de esforços solicit.);
*VETOR_N_SOLIC, VETOR_Mlx_SOLIC, VETOR_Mly_SOLIC: Vetores (N_SECAOx1) que contém
* os esforços de primeira ordem que solicitam as seções transversais do pilar
* (valores de cálculo); convenção de sinais:
*   valores      -> força normal de compressão;
*   positivos:   -> momento definido por uma excentricidade positiva no eixo x;
*               -> momento definido por uma excentricidade positiva no eixo y;
* não há restrições quanto à esses valores, que podem ser positivos, negativos
* ou zero;
*VETOR_N_SOLIC_CONTRAV: No caso de o pilar em análise ser uma estrutura de
* contraventamento (núcleo, pilar parede, etc.), VETOR_N_SOLIC_CONTRAV é um
* vetor (N_SECAOx1) que contém as forças normais totais que agem na estrutura
* contraventada no nível de cada seção transversal do pilar em análise (valores
* de cálculo); convenção de sinal: positivo se for de compressão;
*TIPO_PILAR_X, TIPO_PILAR_Y: Número que determina o tipo de vinculação
* (condições de contorno) do pilar nas direções x e y:
* 1 --> pilar biapoiado
* 2 --> pilar rigidamente engastado na base (seção 1) e livre no topo
* 3 --> pilar elasticamente engastado na base (seção 1) e livre no topo;
*K_MOLA_X, K_MOLA_Y: No caso de pilar do tipo 3, são as constantes de mola
* [FL/RADIANO] nas direções x e y do engaste elástico; k>0;
*Nmax_POLIG_CONCR, Nmax_BARRAS_ACO: Número de nós da poligonal de concreto da
* seção transversal que tiver a poligonal com o maior número de nós; e número
* de barras de aço da seção transversal que tiver a maior quantidade de barras
* de aço; (são os maiores valores dos vetores descritos a seguir);
*VETOR_N_POLIG_CONCR, VETOR_N_BARRAS_ACO: Vetores (N_SECAOx1) que contém o
* número de nós da poligonal de concreto (incluindo o nó de chegada, que é o
* mesmo nó de saída) de cada seção transversal; e contém o número de barras de
* aço de cada seção transversal;
*MATRIZ_X_POLIG_CONCR, MATRIZ_Y_POLIG_CONCR: Matrizes (N_SECAO x
* Nmax_POLIG_CONCR) que contém as coordenadas dos nós da poligonal de concreto
* de cada seção transversal (as poligonais deve ter seus nós numerados no
* sentido anti-horário, com vazios no sentido horário; com o nó de chegada, que
* é o mesmo nó de saída); Exemplo: MATRIZ_X_POLIG_CONCR(3,8) = coordenada x do
* nó 8 da poligonal de concreto que representa a seção transversal 3;
*MATRIZ_X_BARRAS_ACO, MATRIZ_Y_BARRAS_ACO: Matrizes (N_SECAO x Nmax_BARRAS_ACO)
* que contém as coordenadas das barras de aço de cada seção transversal;
* Exemplo: MATRIZ_X_BARRAS_ACO(3,8) = coordenada x da barra de aço 8 da seção
* transversal 3;
*MATRIZ_A_BARRAS_ACO: Matriz (N_SECAO x Nmax_BARRAS_ACO) que contém as áreas das
* barras de aço de cada seção transversal; Exemplo: MATRIZ_A_BARRAS_ACO(3,8) =
* área da barra de aço 8 da seção transversal 3;
*N_m: Número de curvas (polinômios de grau qualquer) que compoem o diagrama

```

```

* tensão-deformação adotado para o concreto; cada curva é definida por um
* somatório do tipo:  $\sigma_c = f_c(C_0.e^0 + C_1.e^1 + C_2.e^2 + \dots + C_m.e^m)$ ; para o
* diagrama parábola-retângulo da NBl/78, temos  $N_m=2$  (curva parabólica e curva
* reta constante);
* VETOR_m: Vetor que contém os graus dos  $N_m$  polinômios; para o parábola-retângulo
* da NBl/78, temos  $VETOR_m=(2,0)$  ( $m=2 \rightarrow$  parábola;  $m=0 \rightarrow$  reta constante);
* MATRIZ_C: Matriz que contém os coeficientes dos  $N_m$  polinômios: linha i contém
* os  $C_k$  ( $k=1$  à  $m+1$ ) coeficientes do polinômio i; para o parábola-retângulo da
* NBl/78, temos: MATRIZ_C = !0 1 -0.25! -> coef. da parábola
* !1 0 0 ! -> coef. da reta constante;
* MATRIZ_e_LIMITES: Matriz que contém as deformações (em por mil) iniciais e
* finais onde age cada um dos  $N_m$  polinômios (deformações estas sem o efeito da
* fluência): linha i contém as deformações iniciais e finais onde age o
* polinômio i; OBS: colocar um número grande para a deformação final do último
* polinômio que compõe o diagrama tensão-deformação adotado para o concreto,
* não finalizando o diagrama no ELU; para o parábola-retângulo da NBl/78, temos:
* MATRIZ_e_LIMITES = !0 2 ! -> def. limites da parábola
* !2 1000! -> def. limites da reta constante;
* fc: Tensão máxima de compressão à que o concreto pode estar sujeito, ou seja,
* é a tensão máxima do diagrama tensão-deformação adotado para o concreto
* (valor de cálculo). Não há consenso quanto ao valor a se utilizar. A NBl/78
* indica  $fc=0.85f_{cd}$  (ELU), mas, para o estudo da estabilidade, alguns autores
* indicam  $fc=0.85f_{ck}$ .
* FI_FLUENCIA: Função de Fluência (Teoria Linear de Fluência);
* fy: Tensão de escoamento do aço (valor de cálculo);
* Es: Módulo de Elasticidade do aço (usualmente igual a 210000MPa);
* CLASSE_ACO: É a classe à que pertence o aço (A ou B);
* VETOR_Wx, VETOR_Wy: Saídas da subrotina: Vetores (N_SECAOx1) que contém os
* deslocamentos nas direções x e y de cada seção transversal;
* VETOR_M2x_SOLIC, VETOR_M2y_SOLIC: Saídas da subrotina: Vetores (N_SECAOx1) que
* contém os momentos de segunda ordem que solicitam as seções transversais do
* pilar (valores de cálculo); convenção de sinais:
* valores -> momento definido por uma excentricidade positiva no eixo x;
* positivos: -> momento definido por uma excentricidade positiva no eixo y;
* ROMPEU: Saída da subrotina: Variável que assume 'NAO' ou 'SIM' conforme o
* pilar seja ou não capaz de suportar os esforços solicitantes dados;
* TIPO_RUPTURA: Saída da subrotina: Variável que assume ' ',
* 'INSTABILIDADE' ou 'ELU RUPTURA' conforme o tipo de ruptura do pilar, caso
* o pilar não suporte os esforços solicitantes dados;
* SECAO_RUPTURA: Saída da subrotina: Número da seção transversal que rompeu, caso
* o pilar não suporte os esforços solicitantes dados.
* -----
* SISTEMA FLEXOR - DIMENSIONAMENTO E VERIFICAÇÃO DE PILARES ESBELTOS DE
* CONCRETO ARMADO COM SEÇÃO QUALQUER SOLICITADOS POR FLEXÃO
* COMPOSTA OBLÍQUA
* DIAG. TENSÃOxDEFORM.DO CONCRETO: POLINÔMIOS DE GRAU QUALQUER
* CONSIDERAÇÃO DA FLUÊNCIA: TEORIA LINEAR DE FLUÊNCIA
* AÇOS BRASILEIROS DO TIPO 'A' OU 'B' COM QUALQUER fy OU Es
* Copyright(c) 1997, Ismael Wilson Cadamuro Júnior
* -----
SUBROUTINE VERIFIC_EXATO (N_SECAO,Z_SECAO,VETOR_N_SOLIC,
+ VETOR_N_SOLIC_CONTRAV,VETOR_M1x_SOLIC,VETOR_M1y_SOLIC,
+ TIPO_PILAR_X,TIPO_PILAR_Y,K_MOLA_X,K_MOLA_Y,Nmax_POLIG_CONCR,
+ VETOR_N_POLIG_CONCR,Nmax_BARRAS_ACO,VETOR_N_BARRAS_ACO,
+ MATRIZ_X_POLIG_CONCR,MATRIZ_Y_POLIG_CONCR,MATRIZ_X_BARRAS_ACO,
+ MATRIZ_Y_BARRAS_ACO,MATRIZ_A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,
+ MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,VETOR_Wx,
+ VETOR_Wy,VETOR_M2x_SOLIC,VETOR_M2y_SOLIC,ROMPEU,TIPO_RUPTURA,
+ SECAO_RUPTURA)
CHARACTER*1 CLASSE_ACO
CHARACTER*3 ROMPEU,ROMPEU2
CHARACTER*13 TIPO_RUPTURA
INTEGER N_SECAO,TIPO_PILAR_X,TIPO_PILAR_Y,Nmax_POLIG_CONCR,
+ VETOR_N_POLIG_CONCR(N_SECAO+1),Nmax_BARRAS_ACO,
+ VETOR_N_BARRAS_ACO(N_SECAO+1),N_m,VETOR_m(N_m+1),SECAO_RUPTURA,
+ N_POLIG_CONCR,N_BARRAS_ACO,N_ITERACOES_MAX,I,K,SECAO,NO,BARRA,
+ N_CALL
INTEGER*2 HOR,MIN,SEG,DSEG
REAL*8 Z_SECAO(N_SECAO+1),VETOR_N_SOLIC(N_SECAO+1),
+ VETOR_N_SOLIC_CONTRAV(N_SECAO+1),VETOR_M1x_SOLIC(N_SECAO+1),
+ VETOR_M1y_SOLIC(N_SECAO+1),K_MOLA_X,K_MOLA_Y,
+ MATRIZ_X_POLIG_CONCR(N_SECAO+1,Nmax_POLIG_CONCR+1),
+ MATRIZ_Y_POLIG_CONCR(N_SECAO+1,Nmax_POLIG_CONCR+1),
+ MATRIZ_X_BARRAS_ACO(N_SECAO+1,Nmax_BARRAS_ACO+1),

```

```

+   MATRIZ_Y_BARRAS_ACO(N_SECAO+1,Nmax_BARRAS_ACO+1),
+   MATRIZ_A_BARRAS_ACO(N_SECAO+1,Nmax_BARRAS_ACO+1),
+   MATRIZ_C(N_m+1,30),MATRIZ_e_LIMITES(N_m+1,2),fc,FI_FLUENCIA,fy,
+   Es,VETOR_Wx(N_SECAO+1),VETOR_Wy(N_SECAO+1),
+   VETOR_M2x_SOLIC(N_SECAO+1),VETOR_M2y_SOLIC(N_SECAO+1),PRECISAO,F
+   ,N_SOLIC,Mx_tot_SOLIC,My_tot_SOLIC,X_POLIG_CONCR[ALLOCATABLE](:)
+   ,Y_POLIG_CONCR[ALLOCATABLE](:),X_BARRAS_ACO[ALLOCATABLE](:),
+   Y_BARRAS_ACO[ALLOCATABLE](:),A_BARRAS_ACO[ALLOCATABLE](:),
+   VETOR_CURVAT_X[ALLOCATABLE](:),VETOR_CURVAT_Y[ALLOCATABLE](:),
+   VETOR_Wxi[ALLOCATABLE](:),VETOR_Wxj[ALLOCATABLE](:),
+   VETOR_Wxk[ALLOCATABLE](:),VETOR_Wyi[ALLOCATABLE](:),
+   VETOR_Wyj[ALLOCATABLE](:),VETOR_Wyk[ALLOCATABLE](:),TETA,ALFA,
+   e_CG,H_ALFA,Nr,Mrx,Mry,TIME1,TIME2,DELTA_TIME,CURVAT_X,CURVAT_Y,
+   Mx_BASE,My_BASE
  ALLOCATE (VETOR_CURVAT_X(N_SECAO+1),VETOR_CURVAT_Y(N_SECAO+1),
+   VETOR_Wxi(N_SECAO+1),VETOR_Wxj(N_SECAO+1),VETOR_Wxk(N_SECAO+1),
+   VETOR_Wyi(N_SECAO+1),VETOR_Wyj(N_SECAO+1),VETOR_Wyk(N_SECAO+1))
*   Obs.: a dimensão dos vetores contém o "+1" para não ocorrer erro no caso
*   de se entrar com nmax_polig_concr ou nmax_barras_aco igual a zero!
  CALL GETTIM(HOR,MIN,SEG,DSEG) !comandos para cálc. do tempo
  TIME1=HOR*60*60+MIN*60+SEG+DSEG/100. ! de execução da subrotina
  TIPO_RUPTURA=' ' !zerando variáveis
  ROMPEU=' '
  ROMPEU2=' '
  SECAO_RUPTURA=0
  N_CALL=0
  N_ITERACOES_MAX=10000 !número máximo de iterações
  PRECISAO=0.001 !precisão de 0.1%
  F=6 !F é utilizado para se verificar a ocorrência da instabilidade
  SECAO=1
  DO WHILE (SECAO.LE.N_SECAO) !zerando variáveis de todas as seções
    VETOR_M2x_SOLIC(SECAO)=0
    VETOR_M2y_SOLIC(SECAO)=0
    VETOR_Wxi(SECAO)=0
    VETOR_Wxj(SECAO)=0
    VETOR_Wxk(SECAO)=0
    VETOR_Wyi(SECAO)=0
    VETOR_Wyj(SECAO)=0
    VETOR_Wyk(SECAO)=0
    SECAO=SECAO+1
  END DO
  K=0
  DO WHILE (2+2.EQ.4) !loop do Método do Equilíbrio
    K=K+1 !'k' é o número da iteração em processamento
    SECAO=1 !loop para cálc. do estado de deformação de todas as seções
    DO WHILE (SECAO.LE.N_SECAO)
      N_POLIG_CONCR=VETOR_N_POLIG_CONCR(SECAO) !preparação das variáveis p/
      N_BARRAS_ACO=VETOR_N_BARRAS_ACO(SECAO) ! para a utilização da sub-
      ALLOCATE (X_POLIG_CONCR(N_POLIG_CONCR+1), ! rotina relacao_m_n_teta
+      Y_POLIG_CONCR(N_POLIG_CONCR+1),X_BARRAS_ACO(N_BARRAS_ACO+1),
+      Y_BARRAS_ACO(N_BARRAS_ACO+1),A_BARRAS_ACO(N_BARRAS_ACO+1))
      NO=1
      DO WHILE (NO.LE.N_POLIG_CONCR)
        X_POLIG_CONCR(NO)=MATRIZ_X_POLIG_CONCR(SECAO,NO)
        Y_POLIG_CONCR(NO)=MATRIZ_Y_POLIG_CONCR(SECAO,NO)
        NO=NO+1
      END DO
      BARRA=1
      DO WHILE (BARRA.LE.N_BARRAS_ACO)
        X_BARRAS_ACO(BARRA)=MATRIZ_X_BARRAS_ACO(SECAO,BARRA)
        Y_BARRAS_ACO(BARRA)=MATRIZ_Y_BARRAS_ACO(SECAO,BARRA)
        A_BARRAS_ACO(BARRA)=MATRIZ_A_BARRAS_ACO(SECAO,BARRA)
        BARRA=BARRA+1
      END DO
      N_SOLIC=VETOR_N_SOLIC(SECAO) !cálc. dos momentos totais
      Mx_tot_SOLIC=VETOR_M1x_SOLIC(SECAO)+VETOR_M2x_SOLIC(SECAO)
      My_tot_SOLIC=VETOR_M1y_SOLIC(SECAO)+VETOR_M2y_SOLIC(SECAO)
      N_CALL=N_CALL+1
      CALL RELACAO_M_N_TETA (N_SOLIC,Mx_tot_SOLIC,My_tot_SOLIC,
+      N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,
+      X_BARRAS_ACO,Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,
+      MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,TETA,ALFA,
+      H_ALFA,e_CG,CURVAT_X,CURVAT_Y,Nr,Mrx,Mry,ROMPEU2)
      DEALLOCATE (X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,

```

```

+       Y_BARRAS_ACO,A_BARRAS_ACO)
      VETOR_CURVAT_X(SECAO)=CURVAT_X
      VETOR_CURVAT_Y(SECAO)=CURVAT_Y
      IF (ROMPEU2.EQ.'SIM') THEN                                !se qualquer seção rompeu...
          ROMPEU='SIM'                                          !                               fim do processo
          TIPO_RUPTURA='ELU RUPTURA '
          SECAO_RUPTURA=SECAO
          GOTO 1
      END IF
      VETOR_Wxi(SECAO)=VETOR_Wxj(SECAO) !deslocamentos 'k' é desta iteração,
      VETOR_Wyi(SECAO)=VETOR_Wyj(SECAO) ! 'j' é da iteração anterior (k-1) e
      VETOR_Wxj(SECAO)=VETOR_Wxk(SECAO) ! 'i' é da penúltima iteração (k-2)
      VETOR_Wyj(SECAO)=VETOR_Wyk(SECAO)
      SECAO=SECAO+1
      END DO!fim do loop para cálc. do estado de deformação de todas as seções
      Mx_BASE=VETOR_M1x_SOLIC(1)+VETOR_M2x_SOLIC(1)!momentos utilizados quando
      My_BASE=VETOR_M1y_SOLIC(1)+VETOR_M2y_SOLIC(1)! o pilar é do tipo 3
      CALL CALC_DESLOCAMENTOS (N_SECAO,Z_SECAO,VETOR_CURVAT_X,!cálc. deslocam.
+       VETOR_CURVAT_Y, TIPO_PILAR_X, TIPO_PILAR_Y, K_MOLA_X, K_MOLA_Y,
+       Mx_BASE, My_BASE, VETOR_Wxk, VETOR_Wyk)
      VETOR_M2x_SOLIC(N_SECAO)=0
      VETOR_M2y_SOLIC(N_SECAO)=0
      SECAO=N_SECAO-1                                          !loop para o cálc. dos momentos de
      DO WHILE (SECAO.GE.1)                                    ! 2ª ordem em todas as seções
          VETOR_M2x_SOLIC(SECAO)=VETOR_M2x_SOLIC(SECAO+1)+
+       (VETOR_N_SOLIC(SECAO+1)+VETOR_N_SOLIC_CONTRAV(SECAO+1))*
+       (VETOR_Wxk(SECAO+1)-VETOR_Wxk(SECAO))
          VETOR_M2y_SOLIC(SECAO)=VETOR_M2y_SOLIC(SECAO+1)+
+       (VETOR_N_SOLIC(SECAO+1)+VETOR_N_SOLIC_CONTRAV(SECAO+1))*
+       (VETOR_Wyk(SECAO+1)-VETOR_Wyk(SECAO))
          SECAO=SECAO-1
      END DO
      IF (K.EQ.1) CYCLE
      I=0
      SECAO=1
      DO WHILE (SECAO.LE.N_SECAO)
          IF (ABS(VETOR_Wxk(SECAO)-VETOR_Wxj(SECAO)).LE. !verifica se o processo
+       ABS(VETOR_Wxj(SECAO)*PRECISAO)+1E-10.AND. ! convergiu para todas
+       ABS(VETOR_Wyk(SECAO)-VETOR_Wyj(SECAO)).LE. ! as seções (o pilar
+       ABS(VETOR_Wyj(SECAO)*PRECISAO)+1E-10) THEN ! é estável)
              I=I+1
          END IF
          SECAO=SECAO+1
      END DO
      IF (I.EQ.N_SECAO) THEN
          ROMPEU='NAO'
          TIPO_RUPTURA=' '
          GOTO 1
      END IF
      IF (K.EQ.2) CYCLE
      SECAO=1
      DO WHILE (SECAO.LE.N_SECAO)
          IF (ABS(SQRT(VETOR_Wxk(SECAO)**2+VETOR_Wyk(SECAO)**2)-
+       SQRT(VETOR_Wxj(SECAO)**2+VETOR_Wyj(SECAO)**2)).GT.
+       F*ABS(SQRT(VETOR_Wxj(SECAO)**2+VETOR_Wyj(SECAO)**2)-
+       SQRT(VETOR_Wxi(SECAO)**2+VETOR_Wyi(SECAO)**2))+1E-5) THEN
              ROMPEU='SIM' !verifica se o processo
              TIPO_RUPTURA='INSTABILIDADE' ! divergiu para alguma
              SECAO_RUPTURA=SECAO ! seção (o pilar é
              GOTO 1 ! instável)
          END IF
          SECAO=SECAO+1
      END DO
      IF (K.GT.N_ITERACOES_MAX) THEN !teste de consistência
          CALL BEEPQQ(1500,1000)
          CALL BEEPQQ(1500,1000)
          WRITE(*,*) 'ERRO (SUBROTINA VERIFIC_EXATO): ',
+       'ULTRAPASSADO O NUMERO MAXIMO DE ITERACOES'
          STOP
      END IF
      END DO
      SECAO=1
      DO WHILE (SECAO.LE.N_SECAO) !os deslocamentos finais são
          VETOR_Wx(SECAO)=VETOR_Wxk(SECAO) ! os da última iteração (k)

```

```

        VETOR_Wy(SECAO)=VETOR_Wyk(SECAO)
        SECAO=SECAO+1
    END DO
*cálculos e testes finais da subrotina:
    CALL GETTIM(HOR,MIN,SEG,DSEG)           !comandos para cálc. do tempo
    TIME2=HOR*60*60+MIN*60+SEG+DSEG/100.   ! de execução da subrotina
    IF (TIME2.LT.TIME1) TIME2=TIME2+24*60*60
    DELTA_TIME=TIME2-TIME1
    WRITE(*,2) K,N_CALL,DELTA_TIME
    2  FORMAT(' VERIFIC_EXATO: ',I4,' iteracoes      ',I4,
+ ' chamadas      ',F8.2,' segundos')      !mensagem informativa
    DEALLOCATE (VETOR_CURVAT_X,VETOR_CURVAT_Y,VETOR_Wxi,VETOR_Wxj,
+ VETOR_Wxk,VETOR_Wyi,VETOR_Wyj,VETOR_Wyk)
    END
*=====fim da subrotina VERIFIC_EXATO=====

```

```

*=====CALC_DESLOCAMENTOS=====

```

```

*Subrotina que, dadas as curvaturas das seções de um pilar, calcula os
* deslocamentos transversais de tais seções. Para isso, efetua uma dupla
* integração da distribuição de curvaturas ao longo do pilar. É assumido que a
* curvatura varia linearmente ao longo de duas seções transversais
* consecutivas, o que implica em variação do tipo parábola do segundo grau para
* as rotações e em variação do tipo parábola cúbica para os deslocamentos
* transversais entre duas seções transversais consecutivas. O pilar deve estar
* referenciado num sistema de coordenadas Oxyz dextrógiro, com seções
* transversais no plano xy e eixo do pilar ao longo do eixo z.
*Unidades de entrada: Sistema de unidades qualquer, desde que consistente;
*N_SECAO: Número de seções transversais do pilar; o pilar deve ser discretizado
* com seções numeradas de 1 a N_SECAO; N_SECAO>=2;
*Z_SECAO: Vetor (N_SECAOx1) que contém as coordenadas z de cada seção
* transversal; a seção 1 deve ter z=0, e as demais z>0; duas seções podem ter o
* mesmo z (caso de pilar com variação brusca de seção ou de esforços solicit.);
*VETOR_CURVAT_X, VETOR_CURVAT_Y: Vetores (N_SECAOx1) que contém as curvaturas
* (1/r) nas direções x e y de cada seção transversal;
*TIPO_PILAR_X, TIPO_PILAR_Y: Número que determina o tipo de vinculação
* (condições de contorno) do pilar nas direções x e y:
* 1 --> pilar biapoiado
* 2 --> pilar rigidamente engastado na base (seção 1) e livre no topo
* 3 --> pilar elasticamente engastado na base (seção 1) e livre no topo;
*K_MOLA_X, K_MOLA_Y: No caso de pilar do tipo 3, são as onstantes de mola
* [FL/RADIANO] nas direções x e y do engaste elástico; k>0;
*Mx_BASE,My_BASE: No caso de pilar do tipo 3, são os momentos fletores que atuam
* na seção da base do pilar (seção do engaste elástico):
* Mx_BASE>0 -> momento definido por uma excentricidade positiva no eixo x
* My_BASE>0 -> momento definido por uma excentricidade positiva no eixo y;
*VETOR_Wx, VETOR_Wy: Saídas da subrotina: Vetores que contém os deslocamentos
* nas direções x e y de cada seção transversal.

```

```

*-----
* SISTEMA FLEXOR - DIMENSIONAMENTO E VERIFICAÇÃO DE PILARES ESBELTOS DE
* CONCRETO ARMADO COM SEÇÃO QUALQUER SOLICITADOS POR FLEXÃO
* COMPOSTA OBLÍQUA
* DIAG. TENSÃOxDEFORM.DO CONCRETO: POLINÔMIOS DE GRAU QUALQUER
* CONSIDERAÇÃO DA FLUÊNCIA: TEORIA LINEAR DE FLUÊNCIA
* AÇOS BRASILEIROS DO TIPO 'A' OU 'B' COM QUALQUER fy OU Es
* Copyright(c) 1997, Ismael Wilson Cadamuro Júnior
*-----

```

```

    SUBROUTINE CALC_DESLOCAMENTOS (N_SECAO,Z_SECAO,VETOR_CURVAT_X,
+ VETOR_CURVAT_Y,TIPO_PILAR_X,TIPO_PILAR_Y,K_MOLA_X,K_MOLA_Y,
+ Mx_BASE,My_BASE,VETOR_Wx,VETOR_Wy)
    INTEGER N_SECAO,TIPO_PILAR_X,TIPO_PILAR_Y,I
    REAL*8 Z_SECAO(N_SECAO+1),VETOR_CURVAT_X(N_SECAO+1),
+ VETOR_CURVAT_Y(N_SECAO+1),K_MOLA_X,K_MOLA_Y,Mx_BASE,My_BASE,
+ VETOR_Wx(N_SECAO+1),VETOR_Wy(N_SECAO+1),ROT1_X[ALLOCATABLE](:),
+ ROT1_Y[ALLOCATABLE](:),Wl_X[ALLOCATABLE](:),Wl_Y[ALLOCATABLE](:)
+ ,C1_X,C1_Y,C2_X,C2_Y
    ALLOCATE (ROT1_X(N_SECAO+1),ROT1_Y(N_SECAO+1),Wl_X(N_SECAO+1),
+ Wl_Y(N_SECAO+1))
* Obs.: a dimensão dos vetores contém o "+1" para não ocorrer erro no caso
* de se entrar com n_secao igual a zero!
    IF (N_SECAO.LT.2) THEN                                     !teste de consistência
        CALL BEEPQQ(1500,1000)
        CALL BEEPQQ(1500,1000)
    END IF

```

```

WRITE(*,*) 'ERRO (SUBROTINA CALC_DESLOCAMENTOS): ',
+ 'PILAR COM MENOS DE 2 SECOES'
STOP
END IF
IF (TIPO_PILAR_X.LT.1.OR.TIPO_PILAR_X.GT.3.OR.      !teste de consistência
+ TIPO_PILAR_Y.LT.1.OR.TIPO_PILAR_Y.GT.3) THEN
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) 'ERRO (SUBROTINA CALC_DESLOCAMENTOS): ',
+ 'PILAR COM VINCULACAO FORA DOS PADROES ACEITOS'
STOP
END IF
ROT1_X(1)=0      !zerando variáveis
ROT1_Y(1)=0
W1_X(1)=0
W1_Y(1)=0
I=2
DO WHILE (I.LE.N_SECAO)      !loop que efetua a dupla integração
IF (Z_SECAO(I).LT.Z_SECAO(I-1)) THEN
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) 'ERRO (SUBROTINA CALC_DESLOCAMENTOS): ', !teste de consis-
+ 'SECOES NUMERADAS NO SENTIDO NEGATIVO DE Z'      !      tência
STOP
END IF      !fórmulas das integrais
ROT1_X(I)=ROT1_X(I-1)+(VETOR_CURVAT_X(I-1)+VETOR_CURVAT_X(I))/2*
+ (Z_SECAO(I)-Z_SECAO(I-1))
ROT1_Y(I)=ROT1_Y(I-1)+(VETOR_CURVAT_Y(I-1)+VETOR_CURVAT_Y(I))/2*
+ (Z_SECAO(I)-Z_SECAO(I-1))
W1_X(I)=W1_X(I-1)+(2*VETOR_CURVAT_X(I-1)+VETOR_CURVAT_X(I))/6*
+ (Z_SECAO(I)-Z_SECAO(I-1))*2+ROT1_X(I-1)*(Z_SECAO(I)-
+ Z_SECAO(I-1))
W1_Y(I)=W1_Y(I-1)+(2*VETOR_CURVAT_Y(I-1)+VETOR_CURVAT_Y(I))/6*
+ (Z_SECAO(I)-Z_SECAO(I-1))*2+ROT1_Y(I-1)*(Z_SECAO(I)-
+ Z_SECAO(I-1))
I=I+1
END DO
C2_X=0
C2_Y=0      !cálc. das constantes de integração
IF (TIPO_PILAR_X.EQ.1) C1_X=-W1_X(N_SECAO)/Z_SECAO(N_SECAO)
IF (TIPO_PILAR_Y.EQ.1) C1_Y=-W1_Y(N_SECAO)/Z_SECAO(N_SECAO)
IF (TIPO_PILAR_X.EQ.2) C1_X=0
IF (TIPO_PILAR_Y.EQ.2) C1_Y=0
IF (TIPO_PILAR_X.EQ.3) C1_X=Mx_BASE/K_MOLA_X
IF (TIPO_PILAR_Y.EQ.3) C1_Y=My_BASE/K_MOLA_Y
I=1
DO WHILE (I.LE.N_SECAO) !aplicação das condições de contorno às integrais
VETOR_Wx(I)=W1_X(I)+C1_X*Z_SECAO(I)+C2_X
VETOR_Wy(I)=W1_Y(I)+C1_Y*Z_SECAO(I)+C2_Y
I=I+1
END DO
DEALLOCATE (ROT1_X,ROT1_Y,W1_X,W1_Y)
END
=====fim da subrotina CALC_DESLOCAMENTOS=====

```

```

=====VERIFIC_PPADRAO=====
*Subrotina que, dado um pilar totalmente definido e seus esforços solicitantes,
* verifica se tal pilar rompe ou não. Para isso, utiliza o Método do Equilíbrio
* com o Processo do Pilar Padrão (veja as características e restrições de tal
* método em FUSCO,P.B.(1981) "Estruturas de Concreto - Solicitações Normais").
* O pilar deve estar referenciado num sistema de coordenadas Oxyz dextrógiro,
* com seções transversais no plano xy e eixo do pilar ao longo do eixo z.
*Unidades de entrada: Sistema de unidades qualquer, desde que consistente;
*N_SOLIC, M1x_SOLIC, Mly_SOLIC:Esforços de primeira ordem que solicitam a seção
* de referência o pilar (valores de cálculo):
* N_SOLIC >0 -> força normal de compressão;
* M1x_SOLIC>0 -> momento definido por uma excentricidade positiva no eixo x;
* Mly_SOLIC>0 -> momento definido por uma excentricidade positiva no eixo y;
* não há restrições quanto à esses valores, que podem ser positivos, negativos
* ou zero;
*N_SOLIC_CONTRAV: No caso de o pilar em análise ser uma estrutura de
* contraventamento (núcleo, pilar parede, etc.), N_SOLIC_CONTRAV é a força

```

```

* normal total que age na estrutura contraventada (valor de cálculo); convenção
* de sinal: positivo se for de compressão;
*L_fl: Comprimento de flambagem do pilar;
*N_POLIG_CONCR e N_BARRAS_ACO: Número de nós da poligonal de concreto (incluindo
* o nó de chegada, que é o mesmo nó de saída) e número de barras de aço da
* seção;
*X_POLIG_CONCR, Y_POLIG_CONCR e X_BARRAS_ACO, Y_BARRAS_ACO: Vetores das
* coordenadas (sistema Oxy, ou seja, não rotacionado) dos nós da poligonal de
* concreto (a poligonal deve ser numerada no sentido anti-horário, com vazios
* no sentido horário; com o nó de chegada, que é o mesmo nó de saída) e das
* coordenadas das barras de aço;
*A_BARRAS_ACO: Vetor das áreas das barras de aço;
*N_m: Número de curvas (polinômios de grau qualquer) que compoem o diagrama
* tensão-deformação adotado para o concreto; cada curva é definida por um
* somatório do tipo:  $\sigma_c = f_c(C_0.e^0 + C_1.e^1 + C_2.e^2 + \dots + C_m.e^m)$ ; para o
* diagrama parábola-retângulo da NBL/78, temos  $N_m=2$  (curva parabólica e curva
* reta constante);
*VETOR_m: Vetor que contém os graus dos  $N_m$  polinômios; para o parábola-retângulo
* da NBL/78, temos  $VETOR_m=(2,0)$  ( $m=2$  -> parábola;  $m=0$  -> reta constante);
*MATRIZ_C: Matriz que contém os coeficientes dos  $N_m$  polinômios: linha  $i$  contém
* os  $C_k$  ( $k=1$  à  $m+1$ ) coeficientes do polinômio  $i$ ; para o parábola-retângulo da
* NBL/78, temos:  $MATRIZ_C = \begin{bmatrix} 1 & 0 & -0.25 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$ 
*  $\rightarrow$  coef. da parábola
*  $\rightarrow$  coef. da reta constante;
*MATRIZ_e_LIMITES: Matriz que contém as deformações (em por mil) iniciais e
* finais onde age cada um dos  $N_m$  polinômios (deformações estas sem o efeito da
* fluência): linha  $i$  contém as deformações iniciais e finais onde age o
* polinômio  $i$ ; OBS: colocar um número grande para a deformação final do último
* polinômio que compõe o diagrama tensão-deformação adotado para o concreto,
* não finalizando o diagrama no ELU; para o parábola-retângulo da NBL/78, temos:
*  $MATRIZ_e_LIMITES = \begin{bmatrix} 1 & 0 & 2 & 0 & 0 \\ 1 & 2 & 1000 & 0 & 0 \end{bmatrix}$ 
*  $\rightarrow$  def. limites da parábola
*  $\rightarrow$  def. limites da reta constante;
*fc: Tensão máxima de compressão à que o concreto pode estar sujeito, ou seja,
* é a tensão máxima do diagrama tensão-deformação adotado para o concreto
* (valor de cálculo). Não há consenso quanto ao valor a se utilizar. A NBL/78
* indica  $fc=0.85f_{cd}$  (ELU), mas, para o estudo da estabilidade, alguns autores
* indicam  $fc=0.85f_{ck}$ .
*FI_FLUENCIA: Função de Fluência (Teoria Linear de Fluência);
*fy: Tensão de escoamento do aço (valor de cálculo);
*Es: Módulo de Elasticidade do aço (usualmente igual a 210000MPa);
*CLASSE_ACO: É a classe à que pertence o aço (A ou B);
*e2x,e2y: Saídas da subrotina: Excentricidades de segunda ordem na seção de
* referência pilar;
*M2x_SOLIC, M2y_SOLIC: Saídas da subrotina: Esforços de segunda ordem que
* solicitam a seção de referência do pilar (valores de cálculo):
*  $M2x\_SOLIC > 0$  -> momento definido por uma excentricidade positiva no eixo x;
*  $M2y\_SOLIC > 0$  -> momento definido por uma excentricidade positiva no eixo y;
*ROMPEU: Saída da subrotina: Variável que assume 'NAO' ou 'SIM' conforme o
* pilar seja ou não capaz de suportar os esforços solicitantes dados;
*TIPO_RUPTURA: Saída da subrotina: Variável que assume ' ',
* 'INSTABILIDADE' ou 'ELU RUPTURA' conforme o tipo de ruptura do pilar, caso
* o pilar não suporte os esforços solicitantes dados;
*
*-----
* SISTEMA FLEXOR - DIMENSIONAMENTO E VERIFICAÇÃO DE PILARES ESBELTOS DE
* CONCRETO ARMADO COM SEÇÃO QUALQUER SOLICITADOS POR FLEXÃO
* COMPOSTA OBLÍQUA
* DIAG. TENSÃOxDEFORM.DO CONCRETO: POLINÔMIOS DE GRAU QUALQUER
* CONSIDERAÇÃO DA FLUÊNCIA: TEORIA LINEAR DE FLUÊNCIA
* AÇOS BRASILEIROS DO TIPO 'A' OU 'B' COM QUALQUER fy OU Es
* Copyright(c) 1997, Ismael Wilson Cadamuro Júnior
*-----
SUBROUTINE VERIFIC_PPADRAO (N_SOLIC,N_SOLIC_CONTRAV,M1x_SOLIC,
+ M1y_SOLIC,L_fl,N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,
+ Y_POLIG_CONCR,X_BARRAS_ACO,Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m
+ ,MATRIZ_C,MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,e2x,
+ e2y,M2x_SOLIC,M2y_SOLIC,ROMPEU,TIPO_RUPTURA)
CHARACTER*1 CLASSE_ACO
CHARACTER*3 ROMPEU,ROMPEU2
CHARACTER*13 TIPO_RUPTURA
INTEGER N_POLIG_CONCR,N_BARRAS_ACO,N_m,VETOR_m(N_m+1),
+ N_ITERACOES_MAX,K
INTEGER*2 HOR,MIN,SEG,DSEG
REAL*8 N_SOLIC,N_SOLIC_CONTRAV,M1x_SOLIC,M1y_SOLIC,M2x_SOLIC,
+ M2y_SOLIC,Mx_tot_SOLIC,My_tot_SOLIC,L_fl,
+ X_POLIG_CONCR(N_POLIG_CONCR+1),Y_POLIG_CONCR(N_POLIG_CONCR+1),

```

```

+ X_BARRAS_ACO(N_BARRAS_ACO+1),Y_BARRAS_ACO(N_BARRAS_ACO+1),
+ A_BARRAS_ACO(N_BARRAS_ACO+1),MATRIZ_C(N_m+1,30),
+ MATRIZ_e_LIMITES(N_m+1,2),fc,FI_FLUENCIA,fy,Es,TETA,ALFA,e_CG,
+ H_ALFA,Nr,Mrx,Mrx,PRECISAO,TIME1,TIME2,DELTA_TIME,CURVAT_X,
+ CURVAT_Y,e2xi,e2xj,e2xk,e2yi,e2yj,e2yk,e2x,e2y,F
*   Obs.: a dimensão dos vetores contém o "+1" para não ocorrer erro no caso
*   de se entrar com n_polig_concr ou n_barras_aco igual a zero!
CALL GETTIM(HOR,MIN,SEG,DSEG)           !comandos para cál. do tempo
TIME1=HOR*60*60+MIN*60+SEG+DSEG/100.   !   de execução da subrotina
IF (N_POLIG_CONCR.LE.3) THEN           !teste de consistência
  CALL BEEPQQ(1500,1000)
  CALL BEEPQQ(1500,1000)
  WRITE(*,*) 'ERRO (SUBROTINA VERIFIC_PPADRAO): ',
+ 'POLIGONAL COM MENOS DE 4 NOS'
  STOP
END IF
IF (X_POLIG_CONCR(1).NE.X_POLIG_CONCR(N_POLIG_CONCR).OR.!teste de consist.
+ Y_POLIG_CONCR(1).NE.Y_POLIG_CONCR(N_POLIG_CONCR)) THEN
  CALL BEEPQQ(1500,1000)
  CALL BEEPQQ(1500,1000)
  WRITE(*,*) 'ERRO (SUBROTINA VERIFIC_PPADRAO): ',
+ 'POLIGONAL COM NOS INICIAL E FINAL COM COORDENADAS DIFERENTES'
  STOP
END IF
TIPO_RUPTURA=' '                       !zerando variáveis
ROMPEU=' '
ROMPEU2=' '
e2xi=0
e2xj=0
e2xk=0
e2yi=0
e2yj=0
e2yk=0
M2x_SOLIC=0
M2y_SOLIC=0
Mx_tot_SOLIC=M1x_SOLIC+M2x_SOLIC
My_tot_SOLIC=M1y_SOLIC+M2y_SOLIC
PRECISAO=0.001                          !precisão de 0.1%
F=6                                       !F é utilizado para se verificar a ocorrência da instabilidade
N_ITERACOES_MAX=10000                   !número máximo de iterações
K=0                                       !'k' é o número da iteração em processamento
DO WHILE (2+2.EQ.4)                      !loop do Método do Equilíbrio
  K=K+1                                  !cál. do estado de deformação da seção de referência
  CALL RELACAO_M_N_TETA (N_SOLIC,Mx_tot_SOLIC,My_tot_SOLIC,
+ N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,
+ X_BARRAS_ACO,Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,
+ MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,TETA,ALFA,
+ H_ALFA,e_CG,CURVAT_X,CURVAT_Y,Nr,Mrx,Mrx,ROMPEU2)
  IF (ROMPEU2.EQ.'SIM') THEN             !se a seção rompeu... fim do processo
    ROMPEU='SIM'
    TIPO_RUPTURA='ELU RUPTURA '
    EXIT
  END IF
  e2xi=e2xj                               !excentricidade 'k' é desta iteração, 'j' é da iteração
  e2yi=e2yj                               ! anterior (k-1) e 'i' é da penúltima iteração (k-2)
  e2xj=e2xk
  e2yj=e2yk
  e2xk=0.1*L_fl**2*CURVAT_X               !cál. das excentricidades nas direções
  e2yk=0.1*L_fl**2*CURVAT_Y               ! 'x' e 'y', para a iteração 'k'
  M2x_SOLIC=(N_SOLIC+N_SOLIC_CONTRAV)*e2xk !cál. dos momentos de 2ª ordem
  M2y_SOLIC=(N_SOLIC+N_SOLIC_CONTRAV)*e2yk
  Mx_tot_SOLIC=M1x_SOLIC+M2x_SOLIC        !cál. dos momentos totais
  My_tot_SOLIC=M1y_SOLIC+M2y_SOLIC
  IF (K.EQ.1) CYCLE
  IF (ABS(e2xk-e2xj).LE.ABS(e2xj*PRECISAO)+1E-15.AND.
+ ABS(e2yk-e2yj).LE.ABS(e2yj*PRECISAO)+1E-15) THEN
    ROMPEU='NAO'                          !verifica se o processo
    TIPO_RUPTURA=' '                     ! convergiu (o pilar
    EXIT                                   ! é estável)
  END IF
  IF (K.EQ.2) CYCLE
  IF (ABS(SQRT(e2xk**2+e2yk**2)-SQRT(e2xj**2+e2yj**2)).GT.F* ! verifica
+ ABS(SQRT(e2xj**2+e2yj**2)-SQRT(e2xi**2+e2yi**2))+1E-5) THEN! se o
  ROMPEU='SIM'                             ! processo

```



```

TIPO_RUPTURA='INSTABILIDADE' ! divergiu
EXIT ! (o pilar é
END IF ! instável)
IF (K.GT.N_ITERACOES_MAX) THEN ! teste de consistência
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) 'ERRO (SUBROTINA VERIFIC_PPADRAO): ',
+ 'ULTRAPASSADO O NUMERO MAXIMO DE ITERACOES'
STOP
END IF
END DO
e2x=e2xk ! as excentricidades finais são as da última iteração (k)
e2y=e2yk
*cálculos e testes finais da subrotina:
CALL GETTIM(HOR,MIN,SEG,DSEG) ! comandos para cálc. do tempo
TIME2=HOR*60*60+MIN*60+SEG+DSEG/100. ! de execução da subrotina
IF (TIME2.LT.TIME1) TIME2=TIME2+24*60*60
DELTA_TIME=TIME2-TIME1
WRITE(*,1) K,DELTA_TIME
1 FORMAT(' VERIFIC_PPADRAO: ',I4,' iteracoes ',
+ ' F8.2,' segundos') ! mensagem informativa
END
*=====fim da subrotina VERIFIC_PPADRAO=====

```

```

*=====SUBROTINA VERIFIC_ELU=====
*Subrotina que, dada uma seção transversal totalmente definida e seus esforços
* solicitantes, verifica se tal seção rompe ou não calculando o máximo momento
* que a seção pode resistir. Para isto, a subrotina calcula o(s) ângulo(s) alfa
* que faz(em) com que o ângulo beta_r seja igual ao beta_s, e que a força
* normal resistente seja igual à força normal solicitante. Admite-se que a
* seção esteja no ELU de ruptura do concreto comprimido ou de deformação
* plástica excessiva da armadura tracionada. O cálculo é feito por tentativas:
* arbitra-se alfa, calcula-se beta_r (subrotina procura_e_CG_teta_ELU) e
* compara-se com beta_s. Para otimizar este processo, esta subrotina utiliza um
* método numérico iterativo de cálculo de zeros de funções: o Método da
* Posição Falsa (ver Ruggiero, M.A.G. e Lopes, V.L.R. - "Cálculo Numérico -
* aspectos teóricos e computacionais" - pg.37). Este método foi escolhido por
* ser eficiente e por ter convergência garantida para o tipo de função em
* questão beta_r=f(alfa), ao contrário de outros métodos mais eficientes, como
* o de Newton-Raphson ou o da Secante. Para o entendimento da subrotina,
* recomenda-se consultar a bibliografia indicada acima. Salienta-se que o
* problema pode ter duas soluções, ou seja, pode haver dois alfa's que
* propiciam beta_r=beta_s (logicamente os módulos de tais momentos serão
* diferentes). Explica-se: se a seção for assimétrica e a força normal
* solicitante for grande, pode ocorrer que o diagrama de interação entre Mrx e
* Mry não contenha em seu interior o ponto (0,0); isto implica em que, dado um
* beta_s, há dois ângulos alfa que propiciam beta_r=beta_s, e, por conseguinte,
* há dois pontos (Mrx,Mry) de mesmo beta_r; neste caso, a análise da ruptura ou
* não da seção deve ser feita comparando-se se o módulo do momento solicitante
* está ou não entre os dois módulos dos momentos resistentes (ou seja, para a
* seção ser considerada não rompida, o módulo do momento fletor solicitante não
* deve ultrapassar um certo valor máximo, mas também não deve ser menor que um
* valor mínimo; caso a seção seja simétrica, o valor mínimo é sempre zero).
* Salienta-se, também, que determinados problemas podem não ter nenhuma
* solução, ou seja, pode ser que não haja nenhum alfa que possibilite
* beta_r=beta_s (ver abaixo explicação da variável EXISTE).
*Unidades de entrada: Sistema de unidades qualquer, desde que consistente;
*N_SOLIC, Mx_SOLIC, My_SOLIC: Esforços que solicitam a seção (valores de
* cálculo):
* N_SOLIC >0 -> força normal de compressão;
* Mx_SOLIC>0 -> momento definido por uma excentricidade positiva no eixo x;
* My_SOLIC>0 -> momento definido por uma excentricidade positiva no eixo y;
* não há restrições quanto à esses valores, que podem ser positivos, negativos
* ou zero;
*N_POLIG_CONCR e N_BARRAS_ACO: Número de nós da poligonal de concreto (incluindo
* o nó de chegada, que é o mesmo nó de saída) e número de barras de aço da
* seção;
*X_POLIG_CONCR, Y_POLIG_CONCR e X_BARRAS_ACO,Y_BARRAS_ACO: Vetores das
* coordenadas (sistema Oxy, ou seja, não rotacionado) dos nós da poligonal de
* concreto (a poligonal deve ser numerada no sentido anti-horário, com vazios
* no sentido horário; com o nó de chegada, que é o mesmo nó de saída) e das
* coordenadas das barras de aço;

```

```

*A_BARRAS_ACO: Vetor das áreas das barras de aço;
*N_m: Número de curvas (polinômios de grau qualquer) que compoem o diagrama
* tensão-deformação adotado para o concreto; cada curva é definida por um
* somatório do tipo:  $\sigma_c = f_c(C_0.e^0 + C_1.e^1 + C_2.e^2 + \dots + C_m.e^m)$ ; para o
* diagrama parábola-retângulo da NBl/78, temos  $N_m=2$  (curva parabólica e curva
* reta constante);
*VETOR_m: Vetor que contém os graus dos  $N_m$  polinômios; para o parábola-retângulo
* da NBl/78, temos  $VETOR_m=(2,0)$  ( $m=2$  -> parábola;  $m=0$  -> reta constante);
*MATRIZ_C: Matriz que contém os coeficientes dos  $N_m$  polinômios: linha i contém
* os  $C_k$  ( $k=1$  à  $m+1$ ) coeficientes do polinômio i; para o parábola-retângulo da
* NBl/78, temos:  $MATRIZ_C = !0 \ 1 \ -0.25!$  -> coef. da parábola
*  $!1 \ 0 \ 0 \ !$  -> coef. da reta constante;
*MATRIZ_e_LIMITES: Matriz que contém as deformações (em por mil) iniciais e
* finais onde age cada um dos  $N_m$  polinômios (deformações estas sem o efeito da
* fluência): linha i contém as deformações iniciais e finais onde age o
* polinômio i; OBS: colocar um número grande para a deformação final do último
* polinômio que compõe o diagrama tensão-deformação adotado para o concreto,
* não finalizando o diagrama no ELU; para o parábola-retângulo da NBl/78, temos:
*  $MATRIZ_e_LIMITES = !0 \ 2 \ !$  -> def. limites da parábola
*  $!2 \ 1000!$  -> def. limites da reta constante;
*fc: Tensão máxima de compressão à que o concreto pode estar sujeito, ou seja,
* é a tensão máxima do diagrama tensão-deformação adotado para o concreto.
* A NBl/78 indica  $fc=0.85fcd$  (ELU).
*FI_FLUENCIA: Função de Fluência (Teoria Linear de Fluência);
*fy: Tensão de escoamento do aço (valor de cálculo);
*Es: Módulo de Elasticidade do aço (usualmente igual a 210000MPa);
*CLASSE_ACO: É a classe à que pertence o aço (A ou B);
*ALFAL,2: Saídas da subrotina: Ângulo de rotação da linha neutra que possibilita
*  $\beta_r$  ser igual ao  $\beta_s$  dado;  $\alpha$  é medido a partir do eixo x no
* sentido trigonométrico (anti-horário), e expresso em radianos;  $\alpha$  pode
* variar de  $-\pi$  à  $+\pi$  radianos; o índice 1 ou 2 refere-se aos problemas com
* duas respostas (conforme explicado acima); caso o problema só admita uma
* resposta,  $\alpha=0$ ;
*H_ALFAL,2: Saídas da subrotina: Altura máxima da poligonal de concreto
* perpendicular à direção da linha neutra ( $\alpha+90$  graus); o índice 1 ou 2
* refere-se aos problemas com duas respostas (conforme explicado acima); caso
* o problema só admita uma resposta,  $h_{\alpha}=0$ ;
*TETA1,2 e e_CG1,2: Saídas da subrotina: Curvatura Majorada Adimensional e
* Deformação do CG que possibilitam que a força normal resistente seja igual à
* força normal solicitante, com a seção atingindo o ELU de ruptura;
*  $teta=1000h_{\alpha}/r$  (na direção perpendicular à da linha neutra, encurtando
* mais as fibras superiores à linha neutra);  $e_{cg}$  é expresso em por mil e
* convencionalmente positivo se for encurtamento; admite-se que o CG esteja
* localizado na origem do sistema de coordenadas: é em relação a este ponto que
* são calculados os esforços resistentes da seção; o índice 1 ou 2 refere-se
* aos problemas com duas respostas (conforme explicado acima); caso o problema
* só admita uma resposta,  $teta=0$  e  $e_{CG}=0$ ;
*Mrx1,2, Mry1,2: Saídas da subrotina: Momentos fletores últimos da seção:
*  $Mrx>0$  --> momento definido por uma excentricidade positiva sobre o eixo x;
*  $Mry>0$  --> momento definido por uma excentricidade positiva sobre o eixo y;
* o índice 1 ou 2 refere-se aos problemas com duas respostas (conforme
* explicado acima); o índice 2 indica o ponto de maior módulo do momento fletor
* resistente, e o 1 o de menor; caso o problema só admita uma resposta,  $Mrx=0$ 
* e  $Mry=0$ ; a direção ( $\beta_r$ ) dos momentos 1 e 2 é a mesma direção do momento
* solicitante dado ( $\beta_s$ );
*EXISTE: Variável que assume 'SIM' ou 'NAO', conforme exista ou não um  $\alpha$  que
* possibilite  $\beta_r$  ser igual ao  $\beta_s$  dado. Explica-se: Se se arbitrar uma
*  $n_{solic}$  grande e a seção for assimétrica em relação aos eixos x e y, pode
* ocorrer que o ângulo  $\beta_r$  se restrinja à uma pequena faixa de variação
* mesmo que se arbitre  $\alpha$  ao longo de todos os 360 graus. Logo, se  $\beta_s$  não
* estiver dentro desta faixa, não haverá  $\alpha$  que iguale  $\beta_r$  ao  $\beta_s$  dado.
* Para exemplificar, suponha uma seção assimétrica submetida à uma  $n_{solic}$ 
* exatamente igual à máxima que a seção pode suportar: só há uma direção para o
* momento resistente ( $\beta_r$ ), qualquer que seja a direção arbitrada para a
* linha neutra; logo, para este caso, não se pode impor um  $\beta_s$  (direção do
* momento solicitante) qualquer, pois não há solução matemática. Se a seção for
* simétrica em relação aos eixos x e y, não há problema: variando-se  $\alpha$  de
* 360 graus, a faixa de variação de  $\beta_r$  também será de 360 graus, e sempre
* haverá solução;
*ROMPEU: Saída da subrotina: Variável que assume 'NAO' ou 'SIM' conforme a seção
* seja ou não capaz de suportar os esforços solicitantes dados;

```

```

* SISTEMA FLEXOR - DIMENSIONAMENTO E VERIFICAÇÃO DE PILARES ESBELTOS DE
* CONCRETO ARMADO COM SEÇÃO QUALQUER SOLICITADOS POR FLEXÃO

```

```

*          COMPOSTA OBLÍQUA
*          DIAG. TENSÃOxDEFORM.DO CONCRETO: POLINÔMIOS DE GRAU QUALQUER
*          CONSIDERAÇÃO DA FLUÊNCIA: TEORIA LINEAR DE FLUÊNCIA
*          AÇOS BRASILEIROS DO TIPO 'A' OU 'B' COM QUALQUER fy OU Es
*          Copyright(c) 1997, Ismael Wilson Cadamuro Júnior
*-----
SUBROUTINE VERIFIC_ELU (N_SOLIC, Mx_SOLIC, My_SOLIC,N_POLIG_CONCR,
+ N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,
+ Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,
+ fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,TETA1,ALFA1,H_ALFA1,e_CG1,Mrx1,
+ Mry1,TETA2,ALFA2,H_ALFA2,e_CG2,Mrx2,Mry2,EXISTE,ROMPEU)
CHARACTER*1 CLASSE_ACO
CHARACTER*3 EXISTE,ROMPEU,SIMETRIA
INTEGER N_POLIG_CONCR,N_BARRAS_ACO,N_m,VETOR_m(N_m+1),N_CALL,
+ N_ITERACOES_MAX,SINAL,I,J,K,ETAPA,ETAPA_AUX,N_RESPOSTA,K_AUX,
+ a_INIC_OK
INTEGER*2 HOR,MIN,SEG,DSEG
REAL*8 BETA_S,N_SOLIC,X_POLIG_CONCR(N_POLIG_CONCR+1),
+ Y_POLIG_CONCR(N_POLIG_CONCR+1),X_BARRAS_ACO(N_BARRAS_ACO+1),
+ Y_BARRAS_ACO(N_BARRAS_ACO+1),A_BARRAS_ACO(N_BARRAS_ACO+1),
+ MATRIZ_C(N_m+1,30),MATRIZ_e_LIMITES(N_m+1,2),fc,FI_FLUENCIA,fy,
+ Es,TETA1,ALFA1,H_ALFA1,e_CG1,Nr1,Mrx1,Mry1,TETA2,ALFA2,H_ALFA2,
+ e_CG2,Nr2,Mrx2,Mry2,TETA,ALFA,H_ALFA,e_CG,Nr,Mrx,Mry,a,b,x,ya,yb
+ ,yx,a0,ya0,M,PRECISAO_x,PRECISAO_y,BETA_R,PI,PASSO,AUX1,AUX2,
+ y_MIN,TIME1,TIME2,DELTA_TIME,AUX3,AUX4,AUX5,AUX6,Nr_MAXIMO,
+ Nr_MINIMO,a1,ya1,b1,yb1,a2,ya2,b2,yb2,Mr1,Mr2,Ms,Mx_SOLIC,
+ My_SOLIC
*          Obs.: a dimensão dos vetores contém o "+1" para não ocorrer erro no caso
*          de se entrar com n_polig_concr ou n_barras_aco igual a zero!
CALL GETTIM(HOR,MIN,SEG,DSEG) !comandos para cálc. do tempo
TIME1=HOR*60*60+MIN*60+SEG+DSEG/100. ! de execução da subrotina
IF (My_SOLIC.NE.0.OR.Mx_SOLIC.NE.0) THEN
  BETA_S=ATAN2(My_SOLIC,Mx_SOLIC)
ELSE
  BETA_S=0
END IF
TETA1=0 !zerando as variáveis
ALFA1=0
H_ALFA1=0
e_CG1=0
Mrx1=0
Mry1=0
TETA2=0
ALFA2=0
H_ALFA2=0
e_CG2=0
Mrx2=0
Mry2=0
ROMPEU=' '
EXISTE='SIM'
PI=ATAN2(0.,-1.)
PRECISAO_x=0.0001 !precisão de 0.0001 rad (0.0057 grau)
PRECISAO_y=0.0001 !precisão de 0.0001 rad (0.0057 grau)
N_ITERACOES_MAX=10000 !número máximo de iterações
N_CALL=0
ETAPA=1 !ETAPA indica o ponto que se está calculando
ETAPA_AUX=1 ! (problemas com mais de uma resposta)
N_RESPOSTA=2
a_INIC_OK=0
IF (BETA_S.LT.-PI) BETA_S=BETA_S+2*PI
IF (BETA_S.GT.+PI) BETA_S=BETA_S-2*PI
CALL COMPRESSAO_TRACAO_UNIFORME_ELU (N_POLIG_CONCR,N_BARRAS_ACO,
+ X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,Y_BARRAS_ACO,
+ A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,fc,
+ FI_FLUENCIA,fy,Es,CLASSE_ACO,AUX1,AUX2,Nr_MAXIMO,AUX3,AUX4,
+ Nr_MINIMO,AUX5,AUX6)
IF (N_SOLIC.LT.Nr_MINIMO.OR.N_SOLIC.GT.Nr_MAXIMO) THEN !teste de consist.
  EXISTE='NAO'
  ROMPEU='SIM'
  GOTO 5
END IF
IF (ABS(AUX3).LE.1E-10.AND.ABS(AUX4).LE.1E-10) THEN !análise da simetria
  SIMETRIA='SIM'
ELSE

```

```

SIMETRIA='NAO'
END IF
IF (SIMETRIA.EQ.'SIM') THEN
  ETAPA=2
  ETAPA_AUX=2
  N_RESPOSTA=1
END IF
*cálc. do 'a' inicial:
a=BETA_S-PI/2 !cálc. do 'a' inicial: direção perpendicular
IF (a.LT.-PI) a=a+2*PI ! à do momento solicitante
IF (a.GT.+PI) a=a-2*PI
ALFA=a
N_CALL=N_CALL+1
CALL PROCURA_e_CG_TETA_ELU (ALFA,N_SOLIC,N_POLIG_CONCR,
+ N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,
+ Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,
+ fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,H_ALFA,TETA,e_CG,Nr,Mrx,Mry)
BETA_R=ATAN2(Mry,Mrx)
ya=BETA_R-BETA_S !cálc. 'ya' inicial
IF (ya.LT.-PI) ya=ya+2*PI
IF (ya.GT.+PI) ya=ya-2*PI
IF (ABS(ya).LE.PRECISAO_y) THEN !testa se 'ya' já não atende à precisão
  a_INIC_OK=1
  ETAPA=2
  ETAPA_AUX=2
  TETA2=TETA
  ALFA2=ALFA
  H_ALFA2=H_ALFA
  e_CG2=e_CG
  Nr2=Nr
  Mrx2=Mrx
  Mry2=Mry
  IF (SIMETRIA.EQ.'SIM') GOTO 1
END IF
*define se o alfa deve aumentar ou diminuir para poder igualar beta_r ao beta_s
*
mais rapidamente:
IF (SIN(BETA_R-BETA_S).LT.0) THEN
  SINAL=+1
ELSE
  SINAL=-1
END IF
IF (SIMETRIA.EQ.'SIM') THEN
  PASSO=SINAL*PI/12 !passo=15 graus na direção oposta à de beta_s
ELSE
  PASSO=SINAL*PI/36 !passo=5 graus na direção oposta à de beta_s
END IF
IF (ETAPA.EQ.2.AND.SIMETRIA.EQ.'NAO') THEN
  a=a+SINAL*PI/360
  ALFA=a
  N_CALL=N_CALL+1
  CALL PROCURA_e_CG_TETA_ELU (ALFA,N_SOLIC,N_POLIG_CONCR,
+ N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,
+ Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,
+ MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,H_ALFA,TETA,
+ e_CG,Nr,Mrx,Mry)
  BETA_R=ATAN2(Mry,Mrx)
  ya=BETA_R-BETA_S !cálc. 'ya' inicial
  IF (ya.LT.-PI) ya=ya+2*PI
  IF (ya.GT.+PI) ya=ya-2*PI
END IF
*cálc. dos 'b' iniciais: Método dos Intervalos Encaixantes
b=a
yb=ya
a0=a
ya0=ya
J=0
DO WHILE (2+2.EQ.4) !cálc. dos 'b' iniciais: Método
  a=b ! dos Intervalos Encaixantes
  ya=yb
  b=b+PASSO
  IF (ABS(b-a0).GT.1.999*PI) THEN !alfa já foi incrementado em 360 graus
  IF (ETAPA.EQ.1) THEN
    EXISTE='NAO' !não encontrou raiz --> não há raiz
    GOTO 1

```

```

ELSE
  IF (a_INIC_OK.EQ.1) GOTO 1                                !só há a raiz 'a' inicial
  N_RESPOSTA=1                                             !encontrou 1 intervalo com raiz
  EXIT
END IF
END IF
ALFA=b
N_CALL=N_CALL+1
CALL PROCURA_e_CG_TETA_ELU (ALFA,N_SOLIC,N_POLIG_CONCR,
+ N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,
+ Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,
+ MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,H_ALFA,TETA,
+ e_CG,Nr,Mrx,Mry)
BETA_R=ATAN2(Mry,Mrx)
yb=BETA_R-BETA_S                                         !cálc. 'yb' inicial
IF (yb.LT.-PI) yb=yb+2*PI
IF (yb.GT.+PI) yb=yb-2*PI
IF (ya*yb.LT.0) THEN!testa se entre 'a' e 'b' há realmente uma raiz: yb
x=(a+b)/2        ! pode mudar de sinal se, com o incremento de alfa,
ALFA=x          ! beta_r for se afastando de beta_s (+ de 180 graus)
N_CALL=N_CALL+1
CALL PROCURA_e_CG_TETA_ELU (ALFA,N_SOLIC,N_POLIG_CONCR,
+ N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,
+ Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,
+ MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,H_ALFA,TETA
+ ,e_CG,Nr,Mrx,Mry)
BETA_R=ATAN2(Mry,Mrx)
yx=BETA_R-BETA_S                                         !cálc. de um ponto entre 'ya' e 'yb' ('yx')
IF (yx.LT.-PI) yx=yx+2*PI
IF (yx.GT.+PI) yx=yx-2*PI
AUX1=yx-ya
IF (AUX1.LT.0) AUX1=AUX1+2*PI
AUX2=yb-ya
IF (AUX2.LT.0) AUX2=AUX2+2*PI
IF (AUX1.LE.AUX2) THEN !cálc. de y_min: o ponto 'yx' está à esquerda
  y_MIN=ya          ! (sentido trigonométrico, ou anti-horário)
ELSE              ! do ponto 'y_min'; logo, para que haja
  y_MIN=yb          ! realmente uma raiz entre 'a' e 'b', é
END IF            ! necessário que o ponto 'y_min' seja o
IF (y_MIN.LE.0) THEN ! ponto negativo; caso contrário, o ponto
  IF (yb*yx.GE.0) THEN ! 'ya' passou para o 'yb' pelo lado oposto
    b=x              ! ao do beta_s (beta_s + 180 graus)
    yb=yx
  ELSE              ! <----- há raiz entre 'a' e 'b'
    a=x
    ya=yx
  END IF
ELSE
  a=b              ! <--- não há raiz entre 'a' e 'b'
  ya=yb
END IF
END IF
IF (ya*yb.LT.0) THEN                                     !encontrou intervalo com raiz:
  IF (ETAPA.EQ.1) THEN
    a2=a
    ya2=ya
    b2=b
    yb2=yb
    ETAPA=2
  ELSE
    a1=a
    ya1=ya
    b1=b
    yb1=yb
    EXIT
  END IF
END IF
J=J+1
IF (J.GT.N_ITERACOES_MAX) THEN                          !teste de consistência
  CALL BEEPQQ(1500,1000)
  CALL BEEPQQ(1500,1000)
  WRITE(*,*) 'ERRO (SUBROTINA VERIFIC_ELU): ULTRAPASSADO ',
+ 'O NUMERO MAXIMO DE ITERACOES'
  STOP

```

```

        END IF
    END DO
*ordena os valores de 'a' e 'b':
    IF (a1.GT.b1) THEN
        AUX1=a1
        AUX2=b1
        a1=AUX2
        b1=AUX1
        AUX1=ya1
        AUX2=yb1
        ya1=AUX2
        yb1=AUX1
    END IF
    IF (a2.GT.b2) THEN
        AUX1=a2
        AUX2=b2
        a2=AUX2
        b2=AUX1
        AUX1=ya2
        AUX2=yb2
        ya2=AUX2
        yb2=AUX1
    END IF
*processo iterativo de cálculo da raiz (Método da Posição Falsa):
    K_AUX=0
    K=0
    ETAPA=1
    IF (ETAPA_AUX.EQ.2) ETAPA=2
3    IF (ETAPA.EQ.1) THEN
        a=a2
        ya=ya2
        b=b2
        yb=yb2
    ELSE
        a=a1
        ya=ya1
        b=b1
        yb=yb1
    END IF
    I=0
    M=ya
    x=a
    yx=ya
    AUX1=1
    DO WHILE (ABS(yx).GT.PRECISAO_Y.OR.K_AUX.EQ.0)
        K_AUX=1
        IF (M*yx.GE.0) THEN
            a=x
            ya=yx
        ELSE
            b=x
            yb=yx
        END IF
        IF (b-a.LE.PRECISAO_X) THEN
            N_CALL=N_CALL+1
            K=K+1
            I=I+1
            x=(a+b)/2
            ALFA=x
            CALL PROCURA_e_CG_TETA_ELU (ALFA,N_SOLIC,N_POLIG_CONCR,
+             N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,
+             Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VECTOR_m,MATRIZ_C,
+             MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,H_ALFA,TETA
+             ,e_CG,Nr,Mrx,Mry)
            BETA_R=ATAN2(Mry,Mrx)
            EXIT
        END IF
        x=(a*yb-b*ya)/(yb-ya)
        IF (I.EQ.8) THEN
            x=(a+b)/2
            I=0
        END IF
        N_CALL=N_CALL+1
        K=K+1
    END WHILE
    !ordenando os valores de 'a' e
    ! de 'b': 'a' deve ser menor
    ! que 'b'
    !contador do número de iterações
    !x fictício para início do loop
    !yx fictício para início do loop
    !testa se b-a atende à precisão
    !a raiz é a média dos dois valores
    !cálc. de 'x'

```

```

I=I+1
ALFA=x
CALL PROCURA_e_CG_TETA_ELU (ALFA,N_SOLIC,N_POLIG_CONCR,
+ N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,
+ Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,
+ MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,H_ALFA,TETA,
+ e_CG,Nr,Mrx,Mry)
BETA_R=ATAN2(Mry,Mrx)
yx=BETA_R-BETA_S !cálc. de 'yx'
IF (yx.LT.-PI) yx=yx+2*PI
IF (yx.GT.+PI) yx=yx-2*PI
IF (K.GT.N_ITERACOES_MAX) THEN !teste de consistência
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) 'ERRO (SUBROTINA VERIFIC_ELU): ULTRAPASSADO ',
+ 'O NUMERO MAXIMO DE ITERACOES'
STOP
END IF
END DO
IF (ETAPA.EQ.1) THEN
TETA2=TETA
ALFA2=ALFA
H_ALFA2=H_ALFA
e_CG2=e_CG
Nr2=Nr
Mrx2=Mrx
Mry2=Mry
a2=a
ya2=ya
b2=b
yb2=yb
ETAPA=2
K_AUX=0
IF (N_RESPOSTA.EQ.2) GOTO 3
ELSE
TETA1=TETA
ALFA1=ALFA
H_ALFA1=H_ALFA
e_CG1=e_CG
Nr1=Nr
Mrx1=Mrx
Mry1=Mry
a1=a
ya1=ya
b1=b
yb1=yb
END IF
*ordena os pontos 1 e 2:(o módulo do momento 1 deve ser menor que o módulo do 2)
1 IF (SQRT(Mrx2**2+Mry2**2).LT.SQRT(Mrx1**2+Mry1**2)) THEN
TETA=TETA1
ALFA=ALFA1
H_ALFA=H_ALFA1
e_CG=e_CG1
Nr=Nr1
Mrx=Mrx1
Mry=Mry1
TETA1=TETA2
ALFA1=ALFA2
H_ALFA1=H_ALFA2
e_CG1=e_CG2
Nr1=Nr2
Mrx1=Mrx2
Mry1=Mry2
TETA2=TETA
ALFA2=ALFA
H_ALFA2=H_ALFA
e_CG2=e_CG
Nr2=Nr
Mrx2=Mrx
Mry2=Mry
END IF
*cálculos e testes finais da subrotina:
CALL GETTIM(HOR,MIN,SEG,DSEG) !comandos para cálc. do tempo
TIME2=HOR*60*60+MIN*60+SEG+DSEG/100. ! de execução da subrotina

```

```

IF (TIME2.LT.TIME1) TIME2=TIME2+24*60*60
DELTA_TIME=TIME2-TIME1
WRITE(*,2) K,N_CALL,DELTA_TIME
2  FORMAT(' VERIFIC_ELU: ',I4,' iteracoes      ',I4,
+  ' chamadas      ',F8.2,' segundos')          !mensagem informativa
IF (ALFA1.LT.-PI) ALFA1=ALFA1+2*PI
IF (ALFA1.GT.+PI) ALFA1=ALFA1-2*PI
IF (ALFA2.LT.-PI) ALFA2=ALFA2+2*PI
IF (ALFA2.GT.+PI) ALFA2=ALFA2-2*PI
ETAPA=1
IF (SQRT(Mrx1**2+Mry1**2).LE.1E-10) ETAPA=2
4  IF (ETAPA.EQ.1) THEN
    a=a1
    ya=ya1
    b=b1
    yb=yb1
    TETA=TETA1
    e_CG=e_CG1
    ALFA=ALFA1
    Mrx=Mrx1
    Mry=Mry1
ELSE
    a=a2
    ya=ya2
    b=b2
    yb=yb2
    TETA=TETA2
    e_CG=e_CG2
    ALFA=ALFA2
    Mrx=Mrx2
    Mry=Mry2
END IF
IF (SQRT(Mrx**2+Mry**2).GE.1E-10) yx=ATAN2(Mry,Mrx)-BETA_S
IF (yx.LT.-PI) yx=yx+2*PI
IF (yx.GT.+PI) yx=yx-2*PI
IF (ABS(yx).GT.4*PRECISAO_y.AND.EXISTE.EQ.'SIM') THEN
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) '-----'
WRITE(*,*) 'AVISO (SUBROTINA VERIFIC_ELU): A SUBROTINA ',
+  ' ACHOU UMA RAIZ EM QUE BETA_R EH DIFERENTE DE BETA_S.'
WRITE(*,*) 'HIPOTEESES:'
WRITE(*,*) ' -PASSO DE 5 GRAUS EH MUITO GRANDE'
WRITE(*,*) ' -PRECISAO_x FOI DETERMINANTE: (b-a) < PRECISAO_x'
WRITE(*,*) ' -IMPRECISAO NUMERICA: LN QUASE PARALELA C/ LADO'
WRITE(*,*) ' -IMPRECISAO NUMERICA: N_SOLIC MUITO GRANDE'
WRITE(*,*) '-----'
WRITE(*,*) 'BETA_R = ',ATAN2(Mry,Mrx)*180/PI,' graus'
WRITE(*,*) 'BETA_S = ',BETA_S*180/PI,' graus'
WRITE(*,*) '-----'
WRITE(*,*) 'a= ',a*180/PI,' grs. -> BETA_R_a=',(ya+BETA_S)*180/PI
WRITE(*,*) 'b= ',b*180/PI,' grs. -> BETA_R_b=',(yb+BETA_S)*180/PI
WRITE(*,*) 'ALFA = ',ALFA*180/PI,' graus'
WRITE(*,*) 'TETA = ',TETA
WRITE(*,*) 'e_CG = ',e_CG,' por mil'
WRITE(*,*) 'Mrx = ',Mrx
WRITE(*,*) 'Mry = ',Mry
WRITE(*,*) '-----'
WRITE(*,*) 'Entre com opcao:'
WRITE(*,*) '      1 --> aceita o ponto assim mesmo'
WRITE(*,*) '      2 --> admite que, para a n_solic arbitrada,'
+  ', nao exista alfa que propicie igualar beta_r a beta_s'
READ(*,*) x
IF (x.EQ.2) EXISTE='NAO'
END IF
IF (ETAPA.EQ.1) THEN
    ETAPA=2
    GOTO 4
END IF
Mr1=SQRT(Mrx1**2+Mry1**2)
Mr2=SQRT(Mrx2**2+Mry2**2)
Ms=SQRT(Mx_SOLIC**2+My_SOLIC**2)
IF (Ms.GE.Mr1.AND.Ms.LE.Mr2.AND.EXISTE.EQ.'SIM') THEN
    ROMPEU='NAO'

```



```

ELSE
ROMPEU='SIM'
END IF
5 END
*=====fim da subrotina VERIFIC_ELU=====

*=====SUBROTINA PROCURA_e_CG_TETA_ELU=====
*Subrotina que, dada a geometria da seção (sist. Oxy), a direção da linha neutra
* e a força normal solicitante, calcula o e_cg (deformação do CG da seção) e o
* teta (curvatura majorada adimensional) que faz com que a força normal
* resistente da seção equilibre a força normal solicitante. Admite-se que a
* seção atinja o ELU de ruptura do concreto comprimido ou de deformação
* plástica excessiva da armadura tracionada. Para este estado de deformação,
* calcula também os esforços resistentes da seção (sist. Oxy).
* O cálculo é feito por tentativas: arbitra-se e_cg e teta (obedecendo-se o ELU
* de ruptura), calcula-se a força normal resistente (subrotina esforcos_secao)
* e compara-se com a força normal solicitante. Para otimizar este processo,
* esta subrotina utiliza um método numérico iterativo de cálculo de zeros de
* funções: o Método da Posição Falsa (ver Ruggiero, M.A.G. e Lopes, V.L.R. -
* "Cálculo Numérico - aspectos teóricos e computacionais" - pg.37). Este método
* foi escolhido por ser eficiente e por ter convergência garantida para o tipo
* de função em questão N_resistente=f(e_CG), ao contrário de outros métodos
* mais eficientes, como o de Newton-Raphson ou o da Secante. Para o
* entendimento da subrotina, recomenda-se consultar a bibliografia indicada
* acima.
*Unidades de entrada: Sistema de unidades qualquer, desde que consistente;
*ALFA: Ângulo (em radianos) de rotação da linha neutra, medido a partir do eixo
* x no sentido anti-horário;
*N_SOLIC: Força normal que solicita a seção; positiva --> compressão;
*N_POLIG_CONCR e N_BARRAS_ACO: Número de nós da poligonal de concreto (incluindo
* o nó de chegada, que é o mesmo nó de saída) e número de barras de aço da
* seção;
*X_POLIG_CONCR, Y_POLIG_CONCR e X_BARRAS_ACO, Y_BARRAS_ACO: Vetores das
* coordenadas (sistema Oxy, ou seja, não rotacionado) dos nós da poligonal de
* concreto (a poligonal deve ser numerada no sentido anti-horário, com vazios
* no sentido horário; com o nó de chegada, que é o mesmo nó de saída) e das
* coordenadas das barras de aço;
*A_BARRAS_ACO: Vetor das áreas das barras de aço;
*N_m: Número de curvas (polinômios de grau qualquer) que compoem o diagrama
* tensão-deformação adotado para o concreto; cada curva é definida por um
* somatório do tipo: sigma_c=fc(C0.e^0 + C1.e^1 + C2.e^2 +...+Cm.e^m); para o
* diagrama parábola-retângulo da NBl/78, temos N_m=2 (curva parabólica e curva
* reta constante);
*VETOR_m: Vetor que contém os graus dos N_m polinômios;para o parábola-retângulo
* da NBl/78, temos VETOR_m=(2,0) (m=2 -> parábola; m=0 -> reta constante);
*MATRIZ_C: Matriz que contém os coeficientes dos N_m polinômios: linha i contém
* os Ck (k=1 à m+1) coeficientes do polinômio i; para o parábola-retângulo da
* NBl/78, temos: MATRIZ_C = !0 1 -0.25! -> coef. da parábola
* !1 0 0 ! -> coef. da reta constante;
*MATRIZ_e_LIMITES: Matriz que contém as deformações (em por mil) iniciais e
* finais onde age cada um dos N_m polinômios (deformações estas sem o efeito da
* fluência): linha i contém as deformações iniciais e finais onde age o
* polinômio i; OBS: colocar um número grande para a deformação final do último
* polinômio que compõe o diagrama tensão-deformação adotado para o concreto,
* não finalizando o diagrama no ELU; para o parábola-retângulo da NBl/78,temos:
* MATRIZ_e_LIMITES = !0 2 ! -> def. limites da parábola
* !2 1000! -> def. limites da reta constante;
*fc: Tensão máxima de compressão à que o concreto pode estar sujeito, ou seja,
* é a tensão máxima do diagrama tensão-deformação adotado para o concreto.
* A NBl/78 indica fc=0.85fcd (ELU);
*FI_FLUENCIA: Função de Fluência (Teoria Linear de Fluência);
*fy: Tensão de escoamento do aço (valor de cálculo);
*Es: Módulo de Elasticidade do aço (usualmente igual a 210000MPa);
*CLASSE_ACO: É a classe à que pertence o aço (A ou B);
*H_ALFA:Saída da subrotina: Altura máxima da poligonal de concreto perpendicular
* à direção da linha neutra (alfa+90 graus);
*TETA e e_CG: Saídas da subrotina: Curvatura Majorada Adimensional e Deformação
* do CG que possibilitam que a força normal resistente seja igual à força
* normal solicitante, com a seção atingindo o ELU de ruptura;
* teta=1000h_alfa/r (na direção perpendicular à da linha neutra, encurtando
* mais as fibras superiores à linha neutra);
* e_cg é expresso em por mil e convencionado positivo se for encurtamento;

```

```

* admite-se que o CG esteja localizado na origem do sistema de coordenadas: é
* em relação a este ponto que são calculados os esforços resistentes da seção;
*Nr, Mrx, Mry: Saídas da subrotina: Esforços resistentes da seção (sist. Oxy):
* Nr >0 --> força normal de compressão;
* Mrx>0 --> momento definido por uma excentricidade positiva sobre o eixo x;
* Mry>0 --> momento definido por uma excentricidade positiva sobre o eixo y;
*-----
* SISTEMA FLEXOR - DIMENSIONAMENTO E VERIFICAÇÃO DE PILARES ESBELTOS DE
* CONCRETO ARMADO COM SEÇÃO QUALQUER SOLICITADOS POR FLEXÃO
* COMPOSTA OBLÍQUA
* DIAG. TENSÃOxDEFORM.DO CONCRETO: POLINÔMIOS DE GRAU QUALQUER
* CONSIDERAÇÃO DA FLUÊNCIA: TEORIA LINEAR DE FLUÊNCIA
* AÇOS BRASILEIROS DO TIPO 'A' OU 'B' COM QUALQUER fy OU Es
* Copyright(c) 1997, Ismael Wilson Cadamuro Júnior
*-----
SUBROUTINE PROCURA_e_CG_TETA_ELU (ALFA,N_SOLIC,N_POLIG_CONCR,
+ N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,
+ Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,
+ fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,H_ALFA,TETA,e_CG,Nr,Mrx,Mry)
CHARACTER*1 CLASSE_ACO
INTEGER N_POLIG_CONCR,N_BARRAS_ACO,N_m,VETOR_m(N_m+1),K,I,N_CALL,
+ N_ITERACOES_MAX
INTEGER*2 HOR,MIN,SEG,DSEG
REAL*8 X_POLIG_CONCR(N_POLIG_CONCR+1),Y_POLIG_CONCR(N_POLIG_CONCR+
+ 1),X_BARRAS_ACO(N_BARRAS_ACO+1),Y_BARRAS_ACO(N_BARRAS_ACO+1),
+ U_POLIG_CONCR[ALLOCATABLE](:),V_POLIG_CONCR[ALLOCATABLE](:),
+ U_BARRAS_ACO[ALLOCATABLE](:),V_BARRAS_ACO[ALLOCATABLE](:),
+ U2_POLIG_CONCR[ALLOCATABLE](:),V2_POLIG_CONCR[ALLOCATABLE](:),
+ U2_BARRAS_ACO[ALLOCATABLE](:),V2_BARRAS_ACO[ALLOCATABLE](:),
+ A_BARRAS_ACO(N_BARRAS_ACO+1),MATRIZ_C(N_m+1,30),
+ MATRIZ_e_LIMITES(N_m+1,2),fc,FI_FLUENCIA,fy,Es,e_CG,Nr,Mrx,Mry,
+ TETA,ALFA,N_SOLIC,Mru,Mrv,e_CG_MINIMO,e_CG_MAXIMO,Nr_MINIMO,
+ Nr_MAXIMO,a,b,ya,yb,yx,M,PRECISAO_x,PRECISAO_y,x,H_ALFA,AUX1,
+ AUX2,TIME1,TIME2,DELTA_TIME,Vcmax,Vsmin,V_CG_NOVO
ALLOCATE (U_POLIG_CONCR(2*N_POLIG_CONCR+1),
+ V_POLIG_CONCR(2*N_POLIG_CONCR+1),U_BARRAS_ACO(N_BARRAS_ACO+1),
+ V_BARRAS_ACO(N_BARRAS_ACO+1),U2_POLIG_CONCR(2*N_POLIG_CONCR+1),
+ V2_POLIG_CONCR(2*N_POLIG_CONCR+1),U2_BARRAS_ACO(N_BARRAS_ACO+1),
+ V2_BARRAS_ACO(N_BARRAS_ACO+1))
* Obs.: a dimensão dos vetores contém o "+1" para não ocorrer erro no caso
* de se entrar com n_polig_concr ou n_barras_aco igual a zero!
CALL GETTIM(HOR,MIN,SEG,DSEG) !comandos para cálc. do tempo
TIME1=HOR*60*60+MIN*60+SEG+DSEG/100. ! de execução da subrotina
N_ITERACOES_MAX=10000 !número máximo de iterações
PRECISAO_x=0.000001 !precisão de 0.000001 por mil
PRECISAO_y=0.0001 !precis.de 0.01% (geralmente é a precisão que predomina)
N_CALL=0
V_CG_NOVO=0
*rotação do sistema de coord. Oxy para o sistema Ouv: (LN paralela ao eixo u)
CALL ROTACAO (ALFA,N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,
+ Y_POLIG_CONCR,X_BARRAS_ACO,Y_BARRAS_ACO,U_POLIG_CONCR,
+ V_POLIG_CONCR,U_BARRAS_ACO,V_BARRAS_ACO)
CALL MAXIMO_MINIMO_DIFEREN(N_POLIG_CONCR,V_POLIG_CONCR,Vcmax,b,
+ H_ALFA)
CALL MAXIMO_MINIMO_DIFEREN(N_BARRAS_ACO,V_BARRAS_ACO,a,Vsmin,b)
*caso necessário, translação do sistema de coord. (é necessário quando a origem
* do sistema não se encontra entre a barra de aço de menor ordenada e o nó de
* maior ordenada da poligonal de concreto menos 3/7 de h_alfa):
IF (Vcmax.LE.3/7.*H_ALFA) V_CG_NOVO=Vsmin/2
IF (Vsmin.GE.0) V_CG_NOVO=Vsmin+H_ALFA/10
a=0
CALL TRANSLACAO (a,V_CG_NOVO,N_POLIG_CONCR,N_BARRAS_ACO,
+ U_POLIG_CONCR,V_POLIG_CONCR,U_BARRAS_ACO,V_BARRAS_ACO,
+ U2_POLIG_CONCR,V2_POLIG_CONCR,U2_BARRAS_ACO,V2_BARRAS_ACO)
*cálc. de 'a' e 'b' iniciais: (e_CG estimados inicialmente com base nos estados
* de deformação de compressão e tração uniformes)
CALL COMPRESSAO_TRACAO_UNIFORME_ELU (N_POLIG_CONCR,N_BARRAS_ACO,
+ U2_POLIG_CONCR,V2_POLIG_CONCR,U2_BARRAS_ACO,V2_BARRAS_ACO,
+ A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,fc,
+ FI_FLUENCIA,fy,Es,CLASSE_ACO,e_CG_MAXIMO,e_CG_MINIMO,Nr_MAXIMO,
+ Mru,Mrv,Nr_MINIMO,AUX1,AUX2)
IF (N_SOLIC.GT.Nr_MAXIMO.OR.N_SOLIC.LT.Nr_MINIMO) THEN !teste de consist.
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)

```

```

WRITE(*,*) 'ERRO (SUBROTINA PROCURA_e_CG_TETA_ELU): ',
+ 'IMPOSSIVEL EQUILIBRAR A FORCA NORMAL SOLICITANTE'
WRITE(*,*) 'Nr_MAXIMO=',Nr_MAXIMO
WRITE(*,*) 'Nr_MINIMO=',Nr_MINIMO
STOP
END IF
a=e_CG_MINIMO           !início da rotina do Método da Posição Falsa:para
b=e_CG_MAXIMO           !      entendimento, ver bibliografia indicada
ya=Nr_MINIMO-N_SOLIC    !cálc. de a,b,ya e yb iniciais
yb=Nr_MAXIMO-N_SOLIC
*teste se os valores iniciais já atendem à precisão:
IF (b-a.LE.PRECISAO_x) THEN           !teste se b-a já não atende à precisão
  N_CALL=N_CALL+1
  e_CG=(a+b)/2                         !a raiz é a média dos dois valores
  CALL e_CGxTETA_ELU (e_CG,N_POLIG_CONCR,N_BARRAS_ACO,
+   V2_POLIG_CONCR,V2_BARRAS_ACO,FI_FLUENCIA,TETA)
  CALL ESFORCOS_SECAO (N_POLIG_CONCR,N_BARRAS_ACO,U2_POLIG_CONCR,
+   V2_POLIG_CONCR,U2_BARRAS_ACO,V2_BARRAS_ACO,A_BARRAS_ACO,N_m,
+   VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,
+   CLASSE_ACO,e_CG,TETA,H_ALFA,Nr,Mru,Mrv)
  GOTO 1
END IF
IF (ABS(ya).LE.ABS(PRECISAO_y*N_SOLIC)+1E-10) THEN           !teste se ya já não
  e_CG=a                                                       ! atende à precisão
  TETA=0
  Nr=Nr_MINIMO
  Mru=AUX1
  Mrv=AUX2
  GOTO 1
END IF
IF (ABS(yb).LE.ABS(PRECISAO_y*N_SOLIC)+1E-10) THEN           !teste se yb já não
  e_CG=b                                                       ! atende à precisão
  TETA=0
  Nr=Nr_MAXIMO
  GOTO 1
END IF
*processo iterativo de cálculo da raiz (Método da Posição Falsa):
I=0
K=0
M=ya
yx=ya                                                         !yx fictício para início do loop
DO WHILE (ABS(yx).GT.ABS(PRECISAO_y*N_SOLIC)+1E-10)
  x=(a*yb-b*ya)/(yb-ya)                                       !cálc. de 'x'
  IF (I.EQ.8) THEN
    x=(a+b)/2
    I=0
  END IF
  K=K+1
  I=I+1
  N_CALL=N_CALL+1
  e_CG=x
  CALL e_CGxTETA_ELU (e_CG,N_POLIG_CONCR,N_BARRAS_ACO,
+   V2_POLIG_CONCR,V2_BARRAS_ACO,FI_FLUENCIA,TETA)
  CALL ESFORCOS_SECAO (N_POLIG_CONCR,N_BARRAS_ACO,U2_POLIG_CONCR,
+   V2_POLIG_CONCR,U2_BARRAS_ACO,V2_BARRAS_ACO,A_BARRAS_ACO,N_m,
+   VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,
+   CLASSE_ACO,e_CG,TETA,H_ALFA,Nr,Mru,Mrv)
  yx=Nr-N_SOLIC                                               !cálc. de 'yx'
  IF (M*yx.GE.0) THEN
    a=x
    ya=yx
  ELSE
    b=x
    yb=yx
  END IF
  IF (b-a.LE.PRECISAO_x) THEN           !testa se b-a atende à precisão
    K=K+1
    I=I+1
    N_CALL=N_CALL+1
    e_CG=(a+b)/2                         !a raiz é a média dos dois valores
    CALL e_CGxTETA_ELU (e_CG,N_POLIG_CONCR,N_BARRAS_ACO,
+   V2_POLIG_CONCR,V2_BARRAS_ACO,FI_FLUENCIA,TETA)
    CALL ESFORCOS_SECAO (N_POLIG_CONCR,N_BARRAS_ACO,U2_POLIG_CONCR
+   ,V2_POLIG_CONCR,U2_BARRAS_ACO,V2_BARRAS_ACO,A_BARRAS_ACO,N_m

```

```

+       ,VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,
+       CLASSE_ACO,e_CG,TETA,H_ALFA,Nr,Mru,Mrv)
      GOTO 1
    END IF
    IF (K.GT.N_ITERACOES_MAX) THEN                                !teste de consistência
      CALL BEEPQQ(1500,1000)
      CALL BEEPQQ(1500,1000)
      WRITE(*,*) 'ERRO (SUBROTINA PROCURA_e_CG_TETA_ELU): ',
+     'ULTRAPASSADO O NUMERO MAXIMO DE ITERACOES'
      STOP
    END IF
  END DO
*cálculos e testes finais da subrotina:
  1  e_CG=e_CG+TETA/H_ALFA*(-V_CG_NOVO) !cálc. de e_cg no sist.não transladado
     Mrv=Mrv+Nr*V_CG_NOVO              !cálc. de Mrv no sist.não transladado
     Mrx=Mru*COS(ALFA)-Mrv*SIN(ALFA)  !cálc.dos esforços no sis.não rotacionado
     Mry=Mru*SIN(ALFA)+Mrv*COS(ALFA)
     CALL GETTIM(HOR,MIN,SEG,DSEG)      !comandos para cálc. do tempo
     TIME2=HOR*60*60+MIN*60+SEG+DSEG/100. ! de execução da subrotina
     IF (TIME2.LT.TIME1) TIME2=TIME2+24*60*60
     DELTA_TIME=TIME2-TIME1
****
     WRITE(*,2) K,N_CALL,DELTA_TIME
  2  FORMAT('   PROCURA_e_CG_TETA_ELU:',I4,' iteracoes ',I4,
+   ' chamadas ',F8.2,' segundos') !mensagem informativa
     DEALLOCATE (U_POLIG_CONCR,V_POLIG_CONCR,U_BARRAS_ACO,V_BARRAS_ACO,
+   U2_POLIG_CONCR,V2_POLIG_CONCR,U2_BARRAS_ACO,V2_BARRAS_ACO)
     IF (ABS(Nr-N_SOLIC).GT.ABS(1.05*PRECISAO_y*N_SOLIC+1E-10)) THEN
       CALL BEEPQQ(1500,1000)
       CALL BEEPQQ(1500,1000)
       WRITE(*,*) '-----'
       WRITE(*,*) 'AVISO (SUBROTINA PROCURA_e_CG_TETA_ELU): A ',
+     'SUBROTINA ACHOU UMA RAIZ EM QUE Nr EH DIFERENTE DE N_SOLIC.'
       WRITE(*,*) 'HIPOTHESES:'
       WRITE(*,*) ' -PRECISAO_x PREDOMINOU SOBRE A PRECISAO_y'
       WRITE(*,*) '-----'
       WRITE(*,*) 'Nr      = ',Nr
       WRITE(*,*) 'N_SOLIC= ',N_SOLIC
       WRITE(*,*) '-----'
       WRITE(*,*) 'Mrx     = ',Mrx
       WRITE(*,*) 'Mry     = ',Mry
       WRITE(*,*) 'TETA    = ',TETA
       WRITE(*,*) 'ALFA    = ',ALFA*180/ATAN2(0.,-1.),' graus'
       WRITE(*,*) 'H_ALFA  = ',H_ALFA
       WRITE(*,*) 'e_CG    = ',e_CG,' por mil'
       WRITE(*,*) '-----'
       WRITE(*,*) 'Entre com opcao:'
       WRITE(*,*) '      1  --> continuar assim mesmo'
       WRITE(*,*) '      --> aperte control+break para parar'
       READ(*,*) I
     END IF
  END
END
*=====fim da subrotina PROCURA_e_CG_TETA_ELU=====

```

```

*=====SUBROTINA e_CGxTETA_ELU=====
*Subrotina que, dado o e_CG ao qual a seção está sujeita, calcula o teta que faz
* com que o estado de deformação da seção atinja o ELU de ruptura do concreto
* comprimido ou de deformação plástica excessiva da armadura tracionada.
* Admite-se que a linha neutra seja paralela ao eixo das abscissas (u).
*ATENÇÃO: Esta subrotina foi confeccionada com base nos domínios de deformação
* definidos na norma NBl/78;
*Unidades de entrada: Sistema de unidades qualquer, desde que consistente;
*e_CG: Deformação (em por mil) do ponto de coordenadas (0,0), onde se admite
* localizar o centro de gravidade da poligonal; é em relação a este ponto que
* são calculados os esforços resistentes da seção; e_cg>0 -> encurtamento;
* o e_CG dado não pode por si só ultrapassar o ELU de ruptura (por exemplo,
* arbitrar e_CG=-11 por mil ou e_CG=+4 por mil!)
*N_POLIG_CONCR e N_BARRAS_ACO: Número de nós da poligonal de concreto (incluindo
* o nó de chegada, que é o mesmo nó de saída) e número de barras de aço da
* seção;
*V_POLIG_CONCR e V_BARRAS_ACO: Vetores das ordenadas dos nós da seção de
* concreto e das barras de aço (sistema uv);
*FI_FLUENCIA: Função de Fluência (Teoria Linear de Fluência);

```

*TETA:Saída da subrotina: Curvatura majorada adimensional ($1000h_{\alpha}/r$), medida
 * na direção perpendicular à da linha neutra (eixo u); teta é sempre positivo,
 * o que significa um maior encurtamento da parte acima da linha neutra em
 * relação à parte abaixo da linha neutra;

 * SISTEMA FLEXOR - DIMENSIONAMENTO E VERIFICAÇÃO DE PILARES ESBELTOS DE
 * CONCRETO ARMADO COM SEÇÃO QUALQUER SOLICITADOS POR FLEXÃO
 * COMPOSTA OBLÍQUA
 * DIAG. TENSÃOxDEFORM.DO CONCRETO: POLINÔMIOS DE GRAU QUALQUER
 * CONSIDERAÇÃO DA FLUÊNCIA: TEORIA LINEAR DE FLUÊNCIA
 * AÇOS BRASILEIROS DO TIPO 'A' OU 'B' COM QUALQUER f_y OU E_s
 * Copyright(c) 1997, Ismael Wilson Cadamuro Júnior

```

SUBROUTINE e_CGxTETA_ELU (e_CG,N_POLIG_CONCR,N_BARRAS_ACO,
+ V_POLIG_CONCR,V_BARRAS_ACO,FI_FLUENCIA,TETA)
INTEGER N_POLIG_CONCR,N_BARRAS_ACO
REAL*8 e_CG,V_POLIG_CONCR(N_POLIG_CONCR+1),
+ V_BARRAS_ACO(N_BARRAS_ACO+1),FI_FLUENCIA,TETA,H_ALFA,V_CONCR_MAX
+ ,V_CONCR_MIN,V_BARRAS_MAX,V_BARRAS_MIN,H_ALFA2,TETA1,TETA2,TETA3
* Obs.: a dimensão dos vetores contém o "+1" para não ocorrer erro no caso
* de se entrar com n_polig_concr ou n_barras_aco igual a zero!
IF (N_POLIG_CONCR.LE.3.AND.N_POLIG_CONCR.NE.0) THEN !teste de consistência
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) 'ERRO (SUBROTINA e_CGxTETA_ELU): POLIGONAL COM',
+ 'MENOS DE 4 NOS'
STOP
END IF
!cálc. h_alfa, v_concr_max e v_barras_min
CALL MAXIMO_MINIMO_DIFEREN (N_POLIG_CONCR,V_POLIG_CONCR,
+ V_CONCR_MAX,V_CONCR_MIN,H_ALFA)
CALL MAXIMO_MINIMO_DIFEREN (N_BARRAS_ACO,V_BARRAS_ACO,
+ V_BARRAS_MAX,V_BARRAS_MIN,H_ALFA2)
IF (N_BARRAS_ACO.EQ.0) V_BARRAS_MIN=V_CONCR_MIN !não há barras de aço
IF (N_POLIG_CONCR.EQ.0) THEN !não há poligonal de concreto
V_CONCR_MAX=V_BARRAS_MAX
H_ALFA=H_ALFA2
END IF
IF (H_ALFA.EQ.0) THEN !teste de consistência
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) 'ERRO (SUBROTINA e_CGxTETA_ELU): ALTURA DA SECAO ',
+ 'IGUAL A ZERO'
STOP
END IF
!cálc. de teta considerando que a seção está na região 1 2 ou 3:
IF (V_CONCR_MAX-3./7*H_ALFA.NE.0) THEN !polo=2 por mil
TETA1=(2*(1+FI_FLUENCIA)-e_CG)*H_ALFA/(V_CONCR_MAX-3./7*H_ALFA)
ELSE
TETA1=1D+300
END IF
IF (V_CONCR_MAX.NE.0) THEN !polo=3.5 por mil
TETA2=(3.5*(1+FI_FLUENCIA)-e_CG)*H_ALFA/(V_CONCR_MAX)
ELSE
TETA2=1D+300
END IF
IF (V_BARRAS_MIN.NE.0) THEN !polo=10 por mil
TETA3=(-10-e_CG)*H_ALFA/(V_BARRAS_MIN)
ELSE
TETA3=1D+300
END IF
IF (TETA1.LT.0) TETA1=1D+300 !teta negativo já pode ser descartado
IF (TETA2.LT.0) TETA2=1D+300
IF (TETA3.LT.0) TETA3=1D+300
TETA=TETA1
IF (TETA2.LT.TETA) TETA=TETA2 !o teta procurado é o menor dos três
IF (TETA3.LT.TETA) TETA=TETA3
IF (TETA.GT.1D299) THEN !teste de consistência
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) 'ERRO (SUBROTINA e_CGxTETA_ELU): e_CG ARBITRADO ',
+ 'NAO PERMITE A OBEDIENCIA AO ELU'
STOP
END IF
END
END
=====fim da subrotina e_CGxTETA_ELU=====

```

```

=====RELACAO_M_N_TETA=====
*Subrotina que, dada uma seção definida geométrica (sist. Oxy) e mecanicamente,
* e seus esforços solicitantes, calcula o estado de deformação (teta, alfa e
* e_cg) que permite aos esforços resistentes equilibrar os esforços
* solicitantes. Para este estado de deformação, calcula também os esforços
* resistentes da seção (sist. Oxy).
* O cálculo é feito por tentativas: arbitra-se teta (Curvatura majorada
* adimensional (1000h_alfa/r), na direção perpendicular à da linha neutra),
* calcula-se M_r (módulo da resultante do momento fletor resistente) através da
* subrotina procura_alfa e compara-se com o M_s (módulo da resultante do
* momento fletor solicitante). Para otimizar este processo, esta subrotina
* utiliza um método numérico iterativo de cálculo de zeros de funções: o
* Método da Secante (ver Ruggiero, M.A.G. e Lopes, V.L.R. - "Cálculo Numérico
* aspectos teóricos e computacionais" - pg.65). Este método foi escolhido por
* ser eficiente e porque não se sabe de início qual o teta máximo ao qual a
* seção pode ser submetida sem ultrapassar o ELU de ruptura do concreto
* comprimido ou de deformação plástica excessiva da armadura tracionada.
* Admitindo-se que a função M_r=f(teta) é convexa (o que é correto na grande
* maioria dos casos), este método têm convergência garantida. Para o caso do
* resultado exigir teta negativo, o método utilizado foi o Método dos
* Intervalos Encaixantes, pois neste caso o Método da Secante teria grande
* probabilidade de divergir. Para o entendimento da subrotina, recomenda-se
* consultar a bibliografia indicada acima.
*Unidades de entrada: Sistema de unidades qualquer, desde que consistente;
*N_SOLIC, Mx_SOLIC, My_SOLIC: Esforços solicitantes da seção (sist. Oxy):
* N_SOLIC >0 -> força normal de compressão;
* Mx_SOLIC>0 -> momento definido por uma excentricidade positiva no eixo x;
* My_SOLIC>0 -> momento definido por uma excentricidade positiva no eixo y;
* não há restrições quanto à esses valores, que podem ser positivos, negativos
* ou zero;
*N_POLIG_CONCR e N_BARRAS_ACO: Número de nós da poligonal de concreto (incluindo
* o nó de chegada, que é o mesmo nó de saída) e número de barras de aço da
* seção;
*X_POLIG_CONCR, Y_POLIG_CONCR e X_BARRAS_ACO, Y_BARRAS_ACO: Vetores das
* coordenadas (sistema Oxy, ou seja, não rotacionado) dos nós da poligonal de
* concreto (a poligonal deve ser numerada no sentido anti-horário, com vazios
* no sentido horário; com o nó de chegada, que é o mesmo nó de saída) e das
* coordenadas das barras de aço;
*A_BARRAS_ACO: Vetor das áreas das barras de aço;
*N_m: Número de curvas (polinômios de grau qualquer) que compoem o diagrama
* tensão-deformação adotado para o concreto; cada curva é definida por um
* somatório do tipo: sigma_c=fc(C0.e^0 + C1.e^1 + C2.e^2 + ...+Cm.e^m); para o
* diagrama parábola-retângulo da NBl/78, temos N_m=2 (curva parabólica e curva
* reta constante);
*VETOR_m: Vetor que contém os graus dos N_m polinômios;para o parábola-retângulo
* da NBl/78, temos VETOR_m=(2,0) (m=2 -> parábola; m=0 -> reta constante);
*MATRIZ_C: Matriz que contém os coeficientes dos N_m polinômios: linha i contém
* os Ck (k=1 à m+1) coeficientes do polinômio i; para o parábola-retângulo da
* NBl/78, temos: MATRIZ_C = !0 1 -0.25! -> coef. da parábola
* !1 0 0 ! -> coef. da reta constante;
*MATRIZ_e_LIMITES: Matriz que contém as deformações (em por mil) iniciais e
* finais onde age cada um dos N_m polinômios (deformações estas sem o efeito da
* fluência): linha i contém as deformações iniciais e finais onde age o
* polinômio i; OBS: colocar um número grande para a deformação final do último
* polinômio que compõe o diagrama tensão-deformação adotado para o concreto,
* não finalizando o diagrama no ELU; para o parábola-retângulo da NBl/78,temos:
* MATRIZ_e_LIMITES = !0 2 ! -> def. limites da parábola
* !2 1000! -> def. limites da reta constante;
*fc: Tensão máxima de compressão à que o concreto pode estar sujeito, ou seja,
* é a tensão máxima do diagrama tensão-deformação adotado para o concreto.
* Não há censo quanto ao valor a se utilizar. A NBl/78 indica fc=0.85fcd
* (ELU), mas, para o estudo da estabilidade, alguns autores indicam fc=0.85fck.
*FI_FLUENCIA: Função de Fluência (Teoria Linear de Fluência);
*fy: Tensão de escoamento do aço (valor de cálculo ou não);
*Es: Módulo de Elasticidade do aço (usualmente igual a 210000MPa);
*CLASSE_ACO: É a classe à que pertence o aço (A ou B);
*TETA:Saída da subrotina: Curvatura majorada adimensional (1000h_alfa/r), medida
* na direção perpendicular à da linha neutra; teta pode ser positivo, negativo
* ou zero; teta positivo significa um maior encurtamento da parte acima da
* linha neutra em relação à parte abaixo da linha neutra (ver definição de
* alfa abaixo);

```

```

*ALFA:Saída da subrotina: Ângulo de rotação da linha neutra que possibilita beta_r
* ser igual à beta_s; alfa é medido a partir do eixo x no sentido
* trigonométrico (anti-horário), e expresso em radianos; alfa pode variar de
* -pi à +pi radianos;
*H_ALFA:Saída da subrotina: Altura máxima da poligonal de concreto perpendicular
* à direção da linha neutra (alfa+90 graus);
*e_CG: Saída da subrotina: Deformação do CG que possibilita que a força normal
* resistente seja igual à força normal solicitante; e_CG é expresso em por mil
* e convencionado positivo se for encurtamento. Admite-se que o CG esteja
* localizado na origem do sistema de coordenadas: é em relação a este ponto que
* são calculados os esforços resistentes da seção;
*CURVAT_X,CURVAT_Y: Saídas da subrotina: Curvaturas dimensionais (1/r) nas
* direções x e y, respectivamente. CURVAT_X=-(TETA/1000h_alfa).SEN(ALFA);
* CURVAT_Y=(TETA/1000h_alfa).COS(ALFA);
*Nr, Mrx, Mry: Saídas da subrotina: Esforços resistentes da seção (sist. Oxy):
* Nr >0 --> força normal de compressão;
* Mrx>0 --> momento definido por uma excentricidade positiva sobre o eixo x;
* Mry>0 --> momento definido por uma excentricidade positiva sobre o eixo y;
*ROMPEU: Variável que assume 'SIM' ou 'NAO' conforme a seção tenha ou não
* superado o ELU de ruptura do concreto comprimido ou de deformação plástica
* excessiva da armadura tracionada.

```

```

-----
* SISTEMA FLEXOR - DIMENSIONAMENTO E VERIFICAÇÃO DE PILARES ESBELTOS DE
* CONCRETO ARMADO COM SEÇÃO QUALQUER SOLICITADOS POR FLEXÃO
* COMPOSTA OBLÍQUA
* DIAG. TENSÃOxDEFORM.DO CONCRETO: POLINÔMIOS DE GRAU QUALQUER
* CONSIDERAÇÃO DA FLUÊNCIA: TEORIA LINEAR DE FLUÊNCIA
* AÇOS BRASILEIROS DO TIPO 'A' OU 'B' COM QUALQUER fy OU Es
* Copyright(c) 1997, Ismael Wilson Cadamuro Júnior
-----

```

```

SUBROUTINE RELACAO_M_N_TETA (N_SOLIC,Mx_SOLIC,My_SOLIC,
+ N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,
+ X_BARRAS_ACO,Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VECTOR_m,MATRIZ_C,
+ MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,TETA,ALFA,
+ H_ALFA,e_CG,CURVAT_X,CURVAT_Y,Nr,Mrx,Mry,ROMPEU)
CHARACTER*1 CLASSE_ACO
CHARACTER*3 SIMETRIA
CHARACTER*3 ROMPEU
CHARACTER*3 EXISTE
INTEGER N_POLIG_CONCR,N_BARRAS_ACO,N_m,VECTOR_m(N_m+1),I,J,K,ETAPA,
+ N_ITERACOES_MAX,N_CALL
INTEGER*2 HOR,MIN,SEG,DSEG
REAL*8 N_SOLIC,Mx_SOLIC,My_SOLIC,X_POLIG_CONCR(N_POLIG_CONCR+1),
+ Y_POLIG_CONCR(N_POLIG_CONCR+1),X_BARRAS_ACO(N_BARRAS_ACO+1),
+ Y_BARRAS_ACO(N_BARRAS_ACO+1),A_BARRAS_ACO(N_BARRAS_ACO+1),
+ MATRIZ_C(N_m+1,30),MATRIZ_e_LIMITES(N_m+1,2),fc,FI_FLUENCIA,fy,
+ Es,TETA,ALFA,e_CG,H_ALFA,Nr,Mrx,Mry,U_POLIG_CONCR[ALLOCATABLE]
+ (:),V_POLIG_CONCR[ALLOCATABLE](:),U_BARRAS_ACO[ALLOCATABLE](:),
+ V_BARRAS_ACO[ALLOCATABLE](:),PRECISAO_x,PRECISAO_y,BETA_S,
+ BETA_SI,Nr_MINIMO,Nr_MAXIMO,e_CG_MINIMO,e_CG_MAXIMO,TETA_MAXIMO,
+ M_S,M_R,x0,x1,x2,y0,y1,y2,PASSO,SINAL,Ac,PI,AUX,AUX1,AUX2,AUX3,
+ AUX4,AUX5,AUX6,TIME1,TIME2,DELTA_TIME,CURVAT_X,CURVAT_Y
ALLOCATE (U_POLIG_CONCR(N_POLIG_CONCR+1),
+ V_POLIG_CONCR(N_POLIG_CONCR+1),U_BARRAS_ACO(N_BARRAS_ACO+1),
+ V_BARRAS_ACO(N_BARRAS_ACO+1))
* Obs.: a dimensão dos vetores contém o "+1" para não ocorrer erro no caso
* de se entrar com n_polig_concr ou n_barras_aco igual a zero!
CALL GETTIM(HOR,MIN,SEG,DSEG) !comandos para cálc. do tempo
TIME1=HOR*60*60+MIN*60+SEG+DSEG/100. ! de execução da subrotina
IF (N_POLIG_CONCR.LE.3) THEN !teste de consistência
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) 'ERRO (SUBROTINA RELACAO_M_N_TETA): ',
+ 'POLIGONAL COM MENOS DE 4 NOS'
STOP
END IF
IF (X_POLIG_CONCR(1).NE.X_POLIG_CONCR(N_POLIG_CONCR).OR.!teste de consist.
+ Y_POLIG_CONCR(1).NE.Y_POLIG_CONCR(N_POLIG_CONCR)) THEN
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) 'ERRO (SUBROTINA RELACAO_M_N_TETA): ',
+ 'POLIGONAL COM NOS INICIAL E FINAL COM COORDENADAS DIFERENTES'
STOP
END IF

```

```

CALL COMPRESSAO_TRACAO_UNIFORME_ELU (N_POLIG_CONCR,N_BARRAS_ACO,
+ X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,Y_BARRAS_ACO,
+ A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,fc,
+ FI_FLUENCIA,fy,Es,CLASSE_ACO,AUX1,AUX2,Nr_MAXIMO,AUX3,AUX4,
+ Nr_MINIMO,AUX5,AUX6)
IF (N_SOLIC.LT.Nr_MINIMO.OR.N_SOLIC.GT.Nr_MAXIMO) THEN
  TETA=0 !testando se a a força normal solicitante
  ALFA=0 ! está ou não dentro da faixa
  e_CG=2 ! em que a seção pode suportar
  H_ALFA=1E-10
  Nr=Nr_MAXIMO
  Mrx=AUX1
  Mry=AUX2
  ROMPEU='SIM'
  GOTO 2
END IF
* teta máximo que se pode arbitrar: Esta subrotina arbitra teta (Método da
* Secante) e o usa para calcular o M_R. Ocorre que se o teta arbitrado for
* muito grande, a subrotina procura_e_cg pode não conseguir encontrar um
* e_cg que possibilite equilibrar a N_solic com a Nr, causando um erro de
* ultrapassagem do número máximo de iterações. Experiências mostraram que um
* teta máximo de 50 é mais que suficiente (observe que este teta ultrapassa
* o teta máximo definido pelos domínios de deformação da NBl/78 (34), mesmo
* com fi_fluencia=2 e com uma só barra de aço no cg da seção):
TETA_MAXIMO=50
ROMPEU='NAO'
N_CALL=0
PRECISAO_x=0.000001 !precisão de 0.000001*1000h/r
PRECISAO_y=0.0001 !precis.de 0.01% (geralm., é a precisão que predomina)
N_ITERACOES_MAX=10000 !número máximo de iterações
PI=ATAN2(0.,-1.)
IF (Mx_SOLIC.NE.0.OR.My_SOLIC.NE.0) THEN
  BETA_S=ATAN2(My_SOLIC,Mx_SOLIC) !ângulo do momento solicitante
  ELSE
  BETA_S=ATAN2(1.,0.)
END IF
BETA_SI=BETA_S
M_S=SQRT(Mx_SOLIC**2+My_SOLIC**2) !módulo do momento solicitante
*analisa a simetria da seção e testa se para teta=0 existe alfa que possibilite
* igualar beta_r à beta_s (aproximação inicial para x0):
x0=0
TETA=x0
ALFA=0
CALL PROCURA_e_CG (TETA,ALFA,N_SOLIC,N_POLIG_CONCR,N_BARRAS_ACO,
+ X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,Y_BARRAS_ACO,
+ A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,fc,
+ FI_FLUENCIA,fy,Es,CLASSE_ACO,H_ALFA,e_CG,Nr,Mrx,Mry)
IF (ABS(Mrx).LE.1E-10.AND.ABS(Mry).LE.1E-10) THEN
  SIMETRIA='SIM'
  EXISTE='SIM'
ELSE
  SIMETRIA='NAO'
  AUX=ATAN2(Mry,Mrx)-BETA_S
  IF (AUX.LT.-PI) AUX=AUX+2*PI
  IF (AUX.GT.+PI) AUX=AUX-2*PI
  IF (ABS(AUX).GT.0.0001) THEN
    EXISTE='NAO'
  ELSE
    EXISTE='SIM'
  END IF
END IF
M_R=SQRT(Mrx**2+Mry**2)
y0=M_R-M_S
IF (ABS(y0).LE.ABS(PRECISAO_y*M_S)+1E-10.AND. !testa se y0 já não
+ EXISTE.EQ.'SIM') THEN ! atende à precisão
  CALL ROTACAO (ALFA,N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,
+ Y_POLIG_CONCR,X_BARRAS_ACO,Y_BARRAS_ACO,U_POLIG_CONCR,
+ V_POLIG_CONCR,U_BARRAS_ACO,V_BARRAS_ACO)
  CALL TETA_eCG_ELU (TETA,N_POLIG_CONCR,N_BARRAS_ACO,V_POLIG_CONCR
+ ,V_BARRAS_ACO,FI_FLUENCIA,e_CG_MINIMO,e_CG_MAXIMO)
  IF (e_CG.GT.e_CG_MAXIMO.OR.e_CG.LT.e_CG_MINIMO) ROMPEU='SIM'
  GOTO 2 !testa se a seção rompeu ou não
END IF
*cálc. da aproximação inicial x0: Método dos Intervalos Encaixantes

```



```

1  PASSO=+0.2                !passo de 0.2 à partir de teta=0 até achar um teta
x0=0                        !                que possibilite fazer beta_r=beta_s
ETAPA=1
J=0                          !Método dos Intervalos Encaixantes:
DO WHILE (EXISTE.EQ.'NAO')  !                loop até achar um teta que
x0=x0+PASSO                 !                possibilite fazer beta_r=beta_s
IF (x0.GT.+TETA_MAXIMO) x0=+TETA_MAXIMO
IF (x0.LT.-TETA_MAXIMO) x0=-TETA_MAXIMO
TETA=x0
N_CALL=N_CALL+1
CALL PROCURA_ALFA (TETA,BETA_S,N_SOLIC,SIMETRIA,N_POLIG_CONCR, !cálc. Mr
+ N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,
+ Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,
+ MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,
+ ALFA,H_ALFA,e_CG,Nr,Mrx,Mry,EXISTE)
M_R=SQRT(Mrx**2+Mry**2)
y0=M_R-M_S
IF (ABS(y0).LE.ABS(PRECISAO_y*M_S)+1E-10.AND.                !testa se y0 já não
+ EXISTE.EQ.'SIM') THEN                                     ! atende à precisão
CALL ROTACAO (ALFA,N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,
+ Y_POLIG_CONCR,X_BARRAS_ACO,Y_BARRAS_ACO,U_POLIG_CONCR,
+ V_POLIG_CONCR,U_BARRAS_ACO,V_BARRAS_ACO)
CALL TETA_eCG_ELU (TETA,N_POLIG_CONCR,N_BARRAS_ACO,
+ V_POLIG_CONCR,V_BARRAS_ACO,FI_FLUENCIA,e_CG_MINIMO,
+ e_CG_MAXIMO)
IF (e_CG.GT.e_CG_MAXIMO.OR.e_CG.LT.e_CG_MINIMO) ROMPEU='SIM'
GOTO 2                                                        !testa se a seção rompeu ou não
END IF
IF (EXISTE.EQ.'SIM'.AND.M_S.LT.M_R.AND.x0.GE.1E-3)THEN
IF (ETAPA.EQ.3) THEN
EXISTE='NAO'                                                !rotina para refinar
x0=x0-PASSO                                                !o processo (diminue
PASSO=+0.001                                               ! o passo até 0.001)
ETAPA=4
END IF
IF (ETAPA.EQ.2) THEN
EXISTE='NAO'
x0=x0-PASSO
PASSO=+0.01
ETAPA=3
END IF
IF (ETAPA.EQ.1) THEN
EXISTE='NAO'
x0=x0-PASSO
PASSO=+0.05
ETAPA=2
END IF
END IF
IF (x0.GT.TETA_MAXIMO) THEN                                !teste de consistência
ROMPEU='SIM'
GOTO 2
END IF
J=J+1
IF (J.GT.N_ITERACOES_MAX) THEN                             !teste de consistência
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) 'ERRO (SUBROTINA RELACAO_M_N_TETA): ',
+ 'ULTRAPASSADO O NUMERO MAXIMO DE ITERACOES'
STOP
END IF
END DO
AUX=x0
AUX1=M_R
AUX2=H_ALFA
*ROTINA ESPECÍFICA PARA RESULTADO QUE EXIGE TETA NEGATIVO (neste caso, não se
* pode garantir a precisão dos esforços resistentes com os solicitantes):
IF (M_S.LT.M_R) THEN
x0=-x0                                                       !cálc. do ponto "espelho" de x0 (-x0)
IF (x0.GT.+TETA_MAXIMO) x0=+TETA_MAXIMO
IF (x0.LT.-TETA_MAXIMO) x0=-TETA_MAXIMO
TETA=x0
N_CALL=N_CALL+1
CALL PROCURA_ALFA (TETA,BETA_S,N_SOLIC,SIMETRIA,N_POLIG_CONCR, !cálc. Mr
+ N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,

```

```

+   Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VECTO_r,MATRIZ_C,
+   MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,
+   ALFA,H_ALFA,e_CG,Nr,Mrx,Mry,EXISTE)
M_R=SQRT(Mrx**2+Mry**2)
y0=M_R-M_S
*   o ponto "espelho" de x0 não existe: há 2 alternativas:
IF (EXISTE.EQ.'NAO') THEN
CALL GEOMETRIA (N_POLIG_CONCR,X_POLIG_CONCR,Y_POLIG_CONCR,Ac,!cálc. Ac
+   AUX3,AUX4)
*   alternativa 1: se o "mi" do ponto +x0 for aproximadamente igual ao "mi"
*   solicitante, a solução está em teta positivo (o passo
*   utilizado no cálc. de +x0 foi muito grande)
IF (ABS((AUX1-M_S)/(Ac*fc*AUX2)).LE.2E-3) THEN
x0=AUX
IF (x0.GT.+TETA_MAXIMO) x0=+TETA_MAXIMO
IF (x0.LT.-TETA_MAXIMO) x0=-TETA_MAXIMO
TETA=x0
N_CALL=N_CALL+1
CALL PROCURA_ALFA (TETA,BETA_S,N_SOLIC,SIMETRIA,
+   N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,
+   X_BARRAS_ACO,Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VECTO_r,
+   MATRIZ_C,MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,
+   ALFA,H_ALFA,e_CG,Nr,Mrx,Mry,EXISTE)
IF (EXISTE.EQ.'NAO') THEN
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) 'ERRO (SUBROTINA RELACAO_M_N_TETA): ',!teste de consist
+   'PROBLEMA DE LOGICA NO TRECHO 3 DE TETA NEGATIVO'
STOP
END IF
CALL ROTACAO (ALFA,N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,
+   Y_POLIG_CONCR,X_BARRAS_ACO,Y_BARRAS_ACO,U_POLIG_CONCR,
+   V_POLIG_CONCR,U_BARRAS_ACO,V_BARRAS_ACO)
CALL TETA_eCG_ELU (TETA,N_POLIG_CONCR,N_BARRAS_ACO,
+   V_POLIG_CONCR,V_BARRAS_ACO,FI_FLUENCIA,e_CG_MINIMO,
+   e_CG_MAXIMO)
IF (e_CG.GT.e_CG_MAXIMO.OR.e_CG.LT.e_CG_MINIMO) ROMPEU='SIM'
GOTO 2
!testa se a seção rompeu ou não
END IF
*   alternativa 2: se o "mi" (+x0) não for próximo ao "mi" solicitante... ramo
*   negativo quase vertical (o Método dos Intervalos
*   Encaixantes não consegue "pegar" a raiz(passo muito grande)
IF (ABS(BETA_S-BETA_SI).GE.5*PI/180) THEN
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) 'ERRO (SUBROTINA RELACAO_M_N_TETA): ',
+   'CURVA M-N-TETA COM RAMO NEGATIVO QUASE VERTICAL, MESMO ',
+   'APOS AUMENTAR BETA_S EM 5 GRAUS'
!teste de consistência
STOP
!solução: uma pequena alteração em beta_s para
END IF
! que o método consiga "pegar" a raiz
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) 'AVISO (SUBROTINA RELACAO_M_N_TETA): ',
+   'CURVA M-N-TETA COM RAMO NEGATIVO QUASE VERTICAL.'
WRITE(*,*) '----> BETA_S=BETA_S+1 grau, e recomeça novamente.'
WRITE(*,*) 'pressione <enter>'
PAUSE
BETA_S=BETA_S+PI/180
IF (BETA_S.LT.-PI) BETA_S=BETA_S+2*PI
IF (BETA_S.GT.+PI) BETA_S=BETA_S-2*PI
EXISTE='NAO'
GOTO 1
END IF
*   o ponto "espelho" -x0 existe, mas o resultado está entre -x0 e +x0 (passo
*   utilizado no cálc. de +x0 foi muito grande): adoto o último teta calculado
*   (ponto -x0)
IF (M_S.GE.M_R) THEN
CALL ROTACAO (ALFA,N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,
+   Y_POLIG_CONCR,X_BARRAS_ACO,Y_BARRAS_ACO,U_POLIG_CONCR,
+   V_POLIG_CONCR,U_BARRAS_ACO,V_BARRAS_ACO)
CALL TETA_eCG_ELU (TETA,N_POLIG_CONCR,N_BARRAS_ACO,
+   V_POLIG_CONCR,V_BARRAS_ACO,FI_FLUENCIA,e_CG_MINIMO,
+   e_CG_MAXIMO)
IF (e_CG.GT.e_CG_MAXIMO.OR.e_CG.LT.e_CG_MINIMO) ROMPEU='SIM'

```

```

        GOTO 2                                !testa se a seção rompeu ou não
    END IF
*   o ponto "espelho" -x0 existe e a raiz está em teta < -x0: Método dos
*   Intervalos Encaixantes
    PASSO=0.1
    x2=x0
    ETAPA=1
    J=0
    DO WHILE (2+2.EQ.4)
        x2=x2-PASSO
        IF (x2.GT.+TETA_MAXIMO) x2=+TETA_MAXIMO
        IF (x2.LT.-TETA_MAXIMO) x2=-TETA_MAXIMO
        TETA=x2
        N_CALL=N_CALL+1                        !cálc. Mr
        CALL PROCURA_ALFA (TETA,BETA_S,N_SOLIC,SIMETRIA,N_POLIG_CONCR,
+       N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,
+       Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VECTOR_m,MATRIZ_C,
+       MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,
+       ALFA,H_ALFA,e_CG,Nr,Mrx,Mry,EXISTE)
        M_R=SQRT(Mrx**2+Mry**2)
        y2=M_R-M_S
        IF (ABS(y2).LE.ABS(PRECISAO_y*M_S)+1E-10.AND.        !testa se x0 já não
+       EXISTE.EQ.'SIM') THEN                                ! atende à precisão
            CALL ROTACAO (ALFA,N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,
+            Y_POLIG_CONCR,X_BARRAS_ACO,Y_BARRAS_ACO,U_POLIG_CONCR,
+            V_POLIG_CONCR,U_BARRAS_ACO,V_BARRAS_ACO)
            CALL TETA_eCG_ELU (TETA,N_POLIG_CONCR,N_BARRAS_ACO,
+            V_POLIG_CONCR,V_BARRAS_ACO,FI_FLUENCIA,e_CG_MINIMO,
+            e_CG_MAXIMO)
            IF (e_CG.GT.e_CG_MAXIMO.OR.e_CG.LT.e_CG_MINIMO) ROMPEU='SIM'
            GOTO 2                                !testa se a seção rompeu ou não
        END IF
        IF (EXISTE.EQ.'NAO') THEN                    !rotina para refinar
            IF (ETAPA.EQ.3) THEN                    !o processo (diminue
+            x2=x2+PASSO                            ! o passo até 0.001)
                IF (x2.GT.+TETA_MAXIMO) x2=+TETA_MAXIMO        ! ou para encontrar
                IF (x2.LT.-TETA_MAXIMO) x2=-TETA_MAXIMO        ! a raiz procurada
                TETA=x2
                N_CALL=N_CALL+1
                CALL PROCURA_ALFA (TETA,BETA_S,N_SOLIC,SIMETRIA,        !cálc. Mr
+                N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,
+                X_BARRAS_ACO,Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VECTOR_m,
+                MATRIZ_C,MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,
+                CLASSE_ACO,ALFA,H_ALFA,e_CG,Nr,Mrx,Mry,EXISTE)
                IF (EXISTE.EQ.'NAO') THEN                !teste de consistência
                    CALL BEEPQQ(1500,1000)
                    CALL BEEPQQ(1500,1000)
                    WRITE(*,*) 'ERRO (SUBROTINA RELACAO_M_N_TETA): ',
+                    'PROBLEMA DE LOGICA NO TRECHO 1 DE TETA NEGATIVO'
                    STOP
                END IF
                CALL ROTACAO (ALFA,N_POLIG_CONCR,N_BARRAS_ACO,
+                X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,Y_BARRAS_ACO,
+                U_POLIG_CONCR,V_POLIG_CONCR,U_BARRAS_ACO,V_BARRAS_ACO)
                CALL TETA_eCG_ELU (TETA,N_POLIG_CONCR,N_BARRAS_ACO,
+                V_POLIG_CONCR,V_BARRAS_ACO,FI_FLUENCIA,e_CG_MINIMO,
+                e_CG_MAXIMO)
                IF (e_CG.GT.e_CG_MAXIMO.OR.e_CG.LT.e_CG_MINIMO)
+                ROMPEU='SIM'
            GOTO 2                                !testa se a seção rompeu ou não
        END IF
        IF (ETAPA.EQ.2) THEN
            x2=x2+PASSO
            PASSO=0.001
            ETAPA=3
        END IF
        IF (ETAPA.EQ.1) THEN
            x2=x2+PASSO
            PASSO=0.01
            ETAPA=2
        END IF
    END IF
    IF (M_S.GT.M_R.AND.EXISTE.EQ.'SIM') THEN
        IF (ETAPA.EQ.3) THEN                        !rotina para refinar

```

```

x2=x2+PASSO/2                                !o processo (diminue
IF (x2.GT.+TETA_MAXIMO) x2=+TETA_MAXIMO      ! o passo até 0.001)
IF (x2.LT.-TETA_MAXIMO) x2=-TETA_MAXIMO      ! ou para encontrar
TETA=x2                                        ! a raiz procurada
N_CALL=N_CALL+1
CALL PROCURA_ALFA (TETA,BETA_S,N_SOLIC,SIMETRIA,      !cálc. Mr
+   N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,
+   X_BARRAS_ACO,Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VECTOR_m,
+   MATRIZ_C,MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,
+   CLASSE_ACO,ALFA,H_ALFA,e_CG,Nr,Mrx,Mry,EXISTE)
IF (EXISTE.EQ.'NAO') THEN                      !teste de consistência
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) 'ERRO (SUBROTINA RELACAO_M_N_TETA): ',
+   'PROBLEMA DE LOGICA NO TRECHO 2 DE TETA NEGATIVO'
STOP
END IF
CALL ROTACAO (ALFA,N_POLIG_CONCR,N_BARRAS_ACO,
+   X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,Y_BARRAS_ACO,
+   U_POLIG_CONCR,V_POLIG_CONCR,U_BARRAS_ACO,V_BARRAS_ACO)
CALL TETA_eCG_ELU (TETA,N_POLIG_CONCR,N_BARRAS_ACO,
+   V_POLIG_CONCR,V_BARRAS_ACO,FI_FLUENCIA,e_CG_MINIMO,
+   e_CG_MAXIMO)
IF (e_CG.GT.e_CG_MAXIMO.OR.e_CG.LT.e_CG_MINIMO)
+   ROMPEU='SIM'
GOTO 2                                          !testa se a seção rompeu ou não
END IF
IF (ETAPA.EQ.2) THEN
x2=x2+PASSO
PASSO=0.001
ETAPA=3
END IF
IF (ETAPA.EQ.1) THEN
x2=x2+PASSO
PASSO=0.01
ETAPA=2
END IF
END IF
J=J+1
IF (J.GT.N_ITERACOES_MAX) THEN                  !teste de consistência
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) 'ERRO (SUBROTINA RELACAO_M_N_TETA): ',
+   'ULTRAPASSADO O NUMERO MAXIMO DE ITERACOES'
STOP
END IF
END DO
END IF
*ROTINA ESPECÍFICA PARA RESULTADO QUE EXIGE TETA POSITIVO:
*   testa se x0 já não ultrapassa o ELU de ruptura:
CALL ROTACAO (ALFA,N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,
+   Y_POLIG_CONCR,X_BARRAS_ACO,Y_BARRAS_ACO,U_POLIG_CONCR,
+   V_POLIG_CONCR,U_BARRAS_ACO,V_BARRAS_ACO)
CALL TETA_eCG_ELU (TETA,N_POLIG_CONCR,N_BARRAS_ACO,V_POLIG_CONCR,
+   V_BARRAS_ACO,FI_FLUENCIA,e_CG_MINIMO,e_CG_MAXIMO)
IF (e_CG.GT.e_CG_MAXIMO.OR.e_CG.LT.e_CG_MINIMO) THEN
ROMPEU='SIM'                                  !testando se para os esforços
GOTO 2                                          ! dados a seção ultrapassa
END IF                                         ! ou não o ELU de ruptura
*cálc. da aproximação inicial x1:
x1=x0+1
IF (x1.GT.+TETA_MAXIMO) x1=+TETA_MAXIMO
IF (x1.LT.-TETA_MAXIMO) x1=-TETA_MAXIMO
TETA=x1
N_CALL=N_CALL+1
CALL PROCURA_ALFA (TETA,BETA_S,N_SOLIC,SIMETRIA,N_POLIG_CONCR,      !cálc. Mr
+   N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,
+   Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VECTOR_m,MATRIZ_C,MATRIZ_e_LIMITES,
+   fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,ALFA,H_ALFA,e_CG,Nr,Mrx,Mry,
+   EXISTE)
M_R=SQRT(Mrx**2+Mry**2)
y1=M_R-M_S
*   testa se M_R ultrapassa M_S (x1 deve ser menor que x2):
J=1

```

```

DO WHILE (M_S.LT.M_R)                                !loop para forçar x1<x2 (raíz
x1=(x0+x1)/2                                          ! fora do intervalo x0,x1)
IF (x1.GT.+TETA_MAXIMO) x1=+TETA_MAXIMO
IF (x1.LT.-TETA_MAXIMO) x1=-TETA_MAXIMO
TETA=x1
N_CALL=N_CALL+1
CALL PROCURA_ALFA (TETA,BETA_S,N_SOLIC,SIMETRIA,N_POLIG_CONCR, !cálc. Mr
+ N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,
+ Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,
+ MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,ALFA,H_ALFA,
+ e_CG,Nr,Mrx,Mry,EXISTE)
M_R=SQRT(Mrx**2+Mry**2)
y1=M_R-M_S
J=J+1
IF (J.GE.N_ITERACOES_MAX) EXIT
END DO
*
testa se y1 já não atende à tolerância:
IF (ABS(y1).LE.ABS(PRECISAO_y*M_S)+1E-10) THEN
CALL ROTACAO (ALFA,N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,
+ Y_POLIG_CONCR,X_BARRAS_ACO,Y_BARRAS_ACO,U_POLIG_CONCR,
+ V_POLIG_CONCR,U_BARRAS_ACO,V_BARRAS_ACO)
CALL TETA_eCG_ELU (TETA,N_POLIG_CONCR,N_BARRAS_ACO,V_POLIG_CONCR
+ ,V_BARRAS_ACO,FI_FLUENCIA,e_CG_MINIMO,e_CG_MAXIMO)
IF (e_CG.GT.e_CG_MAXIMO.OR.e_CG.LT.e_CG_MINIMO) ROMPEU='SIM'
GOTO 2
END IF
*processo iterativo de cálculo da raiz: Método da Secante:
K=0
y2=2*(PRECISAO_y*M_S+5)                               !y2 fictício para início do loop
DO WHILE (ABS(y2).GT.ABS(PRECISAO_y*M_S)+1E-10)
AUX=y1-y0
IF (AUX.GE.0) THEN
SINAL=+1
ELSE
SINAL=-1
END IF
IF (ABS(AUX).LT.1E-20) AUX=SINAL*1E-20
x2=x1-y1*(x1-x0)/AUX
IF (x2.GT.+TETA_MAXIMO) x2=+TETA_MAXIMO
IF (x2.LT.-TETA_MAXIMO) x2=-TETA_MAXIMO
TETA=x2
N_CALL=N_CALL+1
K=K+1
CALL PROCURA_ALFA (TETA,BETA_S,N_SOLIC,SIMETRIA,N_POLIG_CONCR, !cálc. Mr
+ N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,
+ Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,
+ MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,ALFA,H_ALFA,
+ e_CG,Nr,Mrx,Mry,EXISTE)
M_R=SQRT(Mrx**2+Mry**2)
y2=M_R-M_S
*
testa se x2 não ultrapassa o ELU de ruptura:
CALL ROTACAO (ALFA,N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,
+ Y_POLIG_CONCR,X_BARRAS_ACO,Y_BARRAS_ACO,U_POLIG_CONCR,
+ V_POLIG_CONCR,U_BARRAS_ACO,V_BARRAS_ACO)
CALL TETA_eCG_ELU (TETA,N_POLIG_CONCR,N_BARRAS_ACO,V_POLIG_CONCR
+ ,V_BARRAS_ACO,FI_FLUENCIA,e_CG_MINIMO,e_CG_MAXIMO)
IF (e_CG.GT.e_CG_MAXIMO.OR.e_CG.LT.e_CG_MINIMO) THEN
ROMPEU='SIM'                                           !testando se para os esforços
GOTO 2                                                 ! dados a seção ultrapassa
END IF                                               ! ou não o ELU de ruptura
*
testa se x2-x1 não atende à tolerância:
IF (ABS(x2-x1).LE.PRECISAO_x) THEN
GOTO 2
END IF
x0=x1
x1=x2
y0=y1
y1=y2
IF (K.GT.N_ITERACOES_MAX) THEN                          !teste de consistência
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) '-----'
WRITE(*,*) 'ERRO (SUBROTINA RELACAO_M_N_TETA): ',
+ 'ULTRAPASSADO O NUMERO MAXIMO DE ITERACOES'

```

```

WRITE(*,*) 'HIPOTEESES:'
WRITE(*,*) ' -A RESPOSTA ESTA ENTRE UM SALTO DA FUNCAO ',
+ 'BETA_R=F(ALFA)'
WRITE(*,*) ' -PASSO EM x EH MUITO PEQUENO'
WRITE(*,*) ' -FALHA LOGICA DA SUBROTINA'
WRITE(*,*) '-----'
WRITE(*,*) 'M_R = ',M_R
WRITE(*,*) 'M_S = ',M_S
WRITE(*,*) '-----'
WRITE(*,*) 'N_SOLIC= ',N_SOLIC
WRITE(*,*) 'N_RESIS= ',Nr
WRITE(*,*) 'BETA_S = ',BETA_S*180/PI,' graus'
WRITE(*,*) 'BETA_R = ',ATAN2(Mry,Mrx)*180/PI,' graus'
WRITE(*,*) 'TETA = ',TETA
WRITE(*,*) 'ALFA = ',ALFA*180/PI,' graus'
WRITE(*,*) 'e_CG = ',e_CG,' por mil'
WRITE(*,*) 'EXISTE = ',EXISTE
WRITE(*,*) '-----'
WRITE(*,*) 'Entre com opcao:'
WRITE(*,*) ' 1 --> continuar assim mesmo'
WRITE(*,*) ' 235984 --> secao atingiu o ELU de ruptura'
READ(*,*) I
IF (I.EQ.235984) ROMPEU='SIM'
GOTO 2
END IF
END DO
*cálculos e testes finais da subrotina:
2 CURVAT_X=-(TETA/1000/H_ALFA)*SIN(ALFA)
CURVAT_Y=+(TETA/1000/H_ALFA)*COS(ALFA)
CALL GETTIM(HOR,MIN,SEG,DSEG) !comandos para cálc. do tempo
TIME2=HOR*60*60+MIN*60+SEG+DSEG/100. ! de execução da subrotina
IF (TIME2.LT.TIME1) TIME2=TIME2+24*60*60
DELTA_TIME=TIME2-TIME1
WRITE(*,3) K,N_CALL,DELTA_TIME
3 FORMAT(' RELACAO_M_N_TETA: ',I4,' iteracoes ',I4,
+ ' chamadas ',F8.2,' segundos') !mensagem informativa
M_R=SQRT(Mrx**2+Mry**2)
y2=M_R-M_S
IF (ABS(y2).GT.ABS(1.05*PRECISAO_y*M_S+1E-15).AND.ROMPEU.EQ.'NAO')
+ THEN
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) '-----'
WRITE(*,*) 'AVISO (SUBROTINA RELACAO_M_N_TETA): A SUBROTINA ',
+ 'ACHOU UMA RAIZ EM QUE M_R EH DIFERENTE DE M_S.'
WRITE(*,*) 'HIPOTEESES:'
WRITE(*,*) ' -IMPRECISAO NUMERICA: M_S MUITO PEQUENO'
WRITE(*,*) ' -RESPOSTA COM TETA NEGATIVO: METODO DE CALCULO ',
+ 'NAO VERIFICA A PRECISAO'
WRITE(*,*) ' -CURVA M-N-O COM RAMO NEGATIVO QUASE VERTICAL'
WRITE(*,*) ' -PRECISAO_x PREDOMINOU SOBRE A PRECISAO_y'
WRITE(*,*) '-----'
WRITE(*,*) 'M_R = ',M_R
WRITE(*,*) 'M_S = ',M_S
WRITE(*,*) '-----'
WRITE(*,*) 'N_SOLIC= ',N_SOLIC
WRITE(*,*) 'N_RESIS= ',Nr
WRITE(*,*) 'BETA_S = ',BETA_S*180/PI,' graus'
WRITE(*,*) 'BETA_R = ',ATAN2(Mry,Mrx)*180/PI,' graus'
WRITE(*,*) 'TETA = ',TETA
WRITE(*,*) 'ALFA = ',ALFA*180/PI,' graus'
WRITE(*,*) 'e_CG = ',e_CG,' por mil'
WRITE(*,*) 'EXISTE = ',EXISTE
WRITE(*,*) '-----'
WRITE(*,*) 'Entre com opcao:'
WRITE(*,*) ' 1 --> continuar assim mesmo'
WRITE(*,*) ' 235984 --> secao atingiu o ELU de ruptura'
READ(*,*) I
IF (I.EQ.235984) ROMPEU='SIM'
END IF
IF (ROMPEU.EQ.'NAO'.AND.EXISTE.EQ.'NAO') THEN
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) '-----'

```

```

WRITE(*,*) 'ERRO (SUBROTINA RELACAO_M_N_TETA): A SUBROTINA ',
+ 'ACHOU UMA RAIZ EM QUE ROMPEU="NAO" E EXISTE="NAO".'
STOP
END IF
DEALLOCATE (U_POLIG_CONCR,V_POLIG_CONCR,U_BARRAS_ACO,V_BARRAS_ACO)
END
=====fim da subrotina RELACAO_M_N_TETA=====

=====COMPRESSAO_TRACAO_UNIFORME_ELU=====
*Subrotina que, dada uma seção definida geométrica e mecanicamente, calcula os
* esforços resistentes para os casos de compressão e tração uniforme (teta=0),
* admitindo-se que a seção esteja no ELU de ruptura do concreto comprimido ou
* de deformação plástica excessiva da armadura tracionada. Esta subrotina
* utiliza as subrotinas teta_eCG_ELU e esforcos_secao.
*Unidades de entrada: Sistema de unidades qualquer, desde que consistente;
*N_POLIG_CONCR e N_BARRAS_ACO: Número de nós da poligonal de concreto (incluindo
* o nó de chegada, que é o mesmo nó de saída) e número de barras de aço da
* seção;
*U_POLIG_CONCR, V_POLIG_CONCR e U_BARRAS_ACO,V_BARRAS_ACO: Vetores das
* coordenadas dos nós da seção de concreto (a poligonal deve ser numerada no
* sentido anti-horário, com vazios no sentido horário; com o nó de chegada) e
* das coordenadas das barras de aço;
*A_BARRAS_ACO: Vetor das áreas das barras de aço;
*N_m: Número de curvas (polinômios de grau qualquer) que compoem o diagrama
* tensão-deformação adotado para o concreto; cada curva é definida por um
* somatório do tipo: sigma_c=fc(C0.e^0 + C1.e^1 + C2.e^2 +...+Cm.e^m); para o
* diagrama parábola-retângulo da NBl/78, temos N_m=2 (curva parabólica e curva
* reta constante);
*VETOR_m: Vetor que contém os graus dos N_m polinômios;para o parábola-retângulo
* da NBl/78, temos VETOR_m=(2,0) (m=2 -> parábola; m=0 -> reta constante);
*MATRIZ_C: Matriz que contém os coeficientes dos N_m polinômios: linha i contém
* os Ck (k=1 à m+1) coeficientes do polinômio i; para o parábola-retângulo da
* NBl/78, temos: MATRIZ_C = !0 1 -0.25! -> coef. da parábola
* !1 0 0 ! -> coef. da reta constante;
*MATRIZ_e_LIMITES: Matriz que contém as deformações (em por mil) iniciais e
* finais onde age cada um dos N_m polinômios (deformações estas sem o efeito da
* fluência): linha i contém as deformações iniciais e finais onde age o
* polinômio i; OBS: colocar um número grande para a deformação final do último
* polinômio que compõe o diagrama tensão-deformação adotado para o concreto,
* não finalizando o diagrama no ELU; para o parábola-retângulo da NBl/78,temos:
* MATRIZ_e_LIMITES = !0 2 ! -> def. limites da parábola
* !2 1000! -> def. limites da reta constante;
*fc: Tensão máxima de compressão à que o concreto pode estar sujeito, ou seja,
* é a tensão máxima do diagrama tensão-deformação adotado para o concreto.
* Não há consenso quanto ao valor a se utilizar. A NBl/78 indica fc=0.85fcd
* (ELU), mas, para o estudo da estabilidade, alguns autores indicam fc=0.85fck.
*FI_FLUENCIA: Função de Fluência (Teoria Linear de Fluência);
*fy: Tensão de escoamento do aço (valor de cálculo ou não);
*Es: Módulo de Elasticidade do aço (usualmente igual a 210000MPa);
*CLASSE_ACO: É a classe à que pertence o aço (A ou B);
*e_CG_COMPR e e_CG_TRAC: Saídas da subrotina: e_CG para os casos de compressão e
* tração uniformes;
*Nr_COMPR, Mrx_COMPR, Mry_COMPR, Nr_TRAC, Mrx_TRAC, Mry_TRAC: Saídas da
* subrotina: Esforços resistentes da seção para o caso de compressão e tração
* uniformes:
* N >0 --> força normal de compressão;
* Mx>0 --> momento definido por uma excentricidade positiva sobre o eixo x;
* My>0 --> momento definido por uma excentricidade positiva sobre o eixo y.
-----
* SISTEMA FLEXOR - DIMENSIONAMENTO E VERIFICAÇÃO DE PILARES ESBELTOS DE
* CONCRETO ARMADO COM SEÇÃO QUALQUER SOLICITADOS POR FLEXÃO
* COMPOSTA OBLÍQUA
* DIAG. TENSÃOxDEFORM.DO CONCRETO: POLINÔMIOS DE GRAU QUALQUER
* CONSIDERAÇÃO DA FLUÊNCIA: TEORIA LINEAR DE FLUÊNCIA
* AÇOS BRASILEIROS DO TIPO 'A' OU 'B' COM QUALQUER fy OU Es
* Copyright(c) 1997, Ismael Wilson Cadamuro Júnior
-----
SUBROUTINE COMPRESSAO_TRACAO_UNIFORME_ELU (N_POLIG_CONCR,
+ N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,
+ Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,
+ fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,e_CG_COMPR,e_CG_TRAC,Nr_COMPR,
+ Mrx_COMPR,Mry_COMPR,Nr_TRAC,Mrx_TRAC,Mry_TRAC)

```

```

CHARACTER*1 CLASSE_ACO
INTEGER N_POLIG_CONCR,N_BARRAS_ACO,N_m,VETOR_m(N_m+1)
REAL*8 X_POLIG_CONCR(N_POLIG_CONCR+1),Y_POLIG_CONCR(N_POLIG_CONCR+
+ 1),X_BARRAS_ACO(N_BARRAS_ACO+1),Y_BARRAS_ACO(N_BARRAS_ACO+1),
+ A_BARRAS_ACO(N_BARRAS_ACO+1),MATRIZ_C(N_m+1,30),
+ MATRIZ_e_LIMITES(N_m+1,2),fc,FI_FLUENCIA,fy,Es,Nr_COMPR,
+ Mrx_COMPR,Mr_y_COMPR,Nr_TRAC,Mrx_TRAC,Mr_y_TRAC,e_CG_TRAC,
+ e_CG_COMPR,TETA,H
*   Obs.: a dimensão dos vetores contém o "+1" para não ocorrer erro no caso
*   de se entrar com n_polig_concr ou n_barras_aco igual a zero!
TETA=0                                     !estado de deformação uniforme
H=1                                         !h=altura da seção; para teta=0, h não influe no resultado
CALL TETA_eCG_ELU (TETA,N_POLIG_CONCR,N_BARRAS_ACO,Y_POLIG_CONCR,
+ Y_BARRAS_ACO,FI_FLUENCIA,e_CG_TRAC,e_CG_COMPR)
CALL ESFORCOS_SECAO (N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,
+ Y_POLIG_CONCR,X_BARRAS_ACO,Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m
+ ,MATRIZ_C,MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,
+ e_CG_TRAC,TETA,H,Nr_TRAC,Mrx_TRAC,Mr_y_TRAC) !esforços para e_cg_tração
CALL ESFORCOS_SECAO (N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,
+ Y_POLIG_CONCR,X_BARRAS_ACO,Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m
+ ,MATRIZ_C,MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,
+ e_CG_COMPR,TETA,H,Nr_COMPR,Mrx_COMPR,Mr_y_COMPR) !esf. para e_cg_compres.
END
=====fim da subrotina COMPRESSAO_TRACAO_UNIFORME_ELU=====

=====SUBROTINA PROCURA_ALFA=====
*Subrotina que, dada a geometria da seção (sist. Oxy), a curvatura, o ângulo
* beta_s (direção da resultante do momento fletor solicitante) e a força normal
* solicitante, calcula o ângulo alfa (inclinação da linha neutra) e a
* deformação e_cg (deformação do CG da seção) que faz com que o ângulo beta_r
* (direção da resultante do momento fletor resistente) seja igual (a menos de
* uma tolerância) ao beta_s dado, e que a força normal resistente seja igual
* (a menos de uma tolerância) à força normal solicitante. Para este estado de
* deformação, calcula também os esforços resistentes da seção (sist. Oxy).
* O cálculo é feito por tentativas: arbitra-se alfa, calcula-se beta_r
* (subrotina procura_e_cg)e compara-se com beta_s. Para otimizar este processo,
* esta subrotina utiliza um método numérico iterativo de cálculo de zeros de
* funções: o Método da Posição Falsa (ver Ruggiero, M.A.G. e Lopes, V.L.R. -
* "Cálculo Numérico - aspectos teóricos e computacionais" - pg.37). Este método
* foi escolhido por ser eficiente e por ter convergência garantida para o tipo
* de função em questão beta_r=f(alfa), ao contrário de outros métodos mais
* eficientes, como o de Newton-Raphson ou o da Secante. Para o entendimento da
* subrotina, recomenda-se consultar a bibliografia indicada acima.
*Unidades de entrada: Sistema de unidades qualquer, desde que consistente;
*TETA: Curvatura majorada adimensional (1000h_alfa/r) - direção perpendicular
* à da linha neutra; teta pode ser positivo, negativo ou zero; teta positivo
* significa um maior encurtamento da parte acima da linha neutra em relação à
* parte abaixo da linha neutra (ver definição de alfa abaixo);
*BETA_S: Ângulo (em radianos) entre o eixo x e a direção da resultante do
* momento fletor solicitante, medido no sentido trigonométrico (anti-horário):
* beta_s = arctg (Msy/Msx); beta_s deve estar entre -pi e +pi;
*N_SOLIC: Força normal que solicita a seção; positiva --> compressão;
*SIMETRIA: Variável que deve assumir 'sim' ou 'nao' conforme a seção seja ou não
* simétrica em relação aos eixos x e y; a simetria deve ser tanto na seção de
* concreto quanto na armadura; a simetria não se refere ao plano de ação do
* momento solicitante, ou seja, a seção pode ser simétrica mas estar submetida
* à flexão composta oblíqua;
*N_POLIG_CONCR e N_BARRAS_ACO: Número de nós da poligonal de concreto (incluindo
* o nó de chegada, que é o mesmo nó de saída) e número de barras de aço da
* seção;
*X_POLIG_CONCR, Y_POLIG_CONCR e X_BARRAS_ACO,Y_BARRAS_ACO: Vetores das
* coordenadas (sistema Oxy, ou seja, não rotacionado) dos nós da poligonal de
* concreto (a poligonal deve ser numerada no sentido anti-horário, com vazios
* no sentido horário; com o nó de chegada, que é o mesmo nó de saída) e das
* coordenadas das barras de aço;
*A_BARRAS_ACO: Vetor das áreas das barras de aço;
*N_m: Número de curvas (polinômios de grau qualquer) que compoem o diagrama
* tensão-deformação adotado para o concreto; cada curva é definida por um
* somatório do tipo: sigma_c=fc(C0.e^0 + C1.e^1 + C2.e^2 +...+Cm.e^m); para o
* diagrama parábola-retângulo da NB1/78, temos N_m=2 (curva parabólica e curva
* reta constante);
*VETOR_m: Vetor que contém os graus dos N_m polinômios;para o parábola-retângulo

```



```

* da NBl/78, temos VETOR_m=(2,0) (m=2 -> parábola; m=0 -> reta constante);
*MATRIZ_C: Matriz que contém os coeficientes dos N_m polinômios: linha i contém
* os Ck (k=1 à m+1) coeficientes do polinômio i; para o parábola-retângulo da
* NBl/78, temos: MATRIZ_C = !0 1 -0.25! -> coef. da parábola
* !1 0 0 ! -> coef. da reta constante;
*MATRIZ_e_LIMITES: Matriz que contém as deformações (em por mil) iniciais e
* finais onde age cada um dos N_m polinômios (deformações estas sem o efeito da
* fluência): linha i contém as deformações iniciais e finais onde age o
* polinômio i; OBS: colocar um número grande para a deformação final do último
* polinômio que compõe o diagrama tensão-deformação adotado para o concreto,
* não finalizando o diagrama no ELU; para o parábola-retângulo da NBl/78, temos:
* MATRIZ_e_LIMITES = !0 2 ! -> def. limites da parábola
* !2 1000! -> def. limites da reta constante;
*fc: Tensão máxima de compressão à que o concreto pode estar sujeito, ou seja,
* é a tensão máxima do diagrama tensão-deformação adotado para o concreto.
* Não há consenso quanto ao valor a se utilizar. A NBl/78 indica fc=0.85fcd
* (ELU), mas, para o estudo da estabilidade, alguns autores indicam fc=0.85fck.
*FI_FLUENCIA: Função de Fluência (Teoria Linear de Fluência);
*fy: Tensão de escoamento do aço (valor de cálculo ou não);
*Es: Módulo de Elasticidade do aço (usualmente igual a 210000MPa);
*CLASSE_ACO: É a classe à que pertence o aço (A ou B);
*ALFA:Saída da subrotina: Ângulo de rotação da linha neutra que possibilita beta_r
* ser igual ao beta_s dado; alfa é medido a partir do eixo x no sentido
* trigonométrico (anti-horário), e expresso em radianos; alfa pode variar de
* -pi à +pi radianos;
*H_ALFA:Saída da subrotina: Altura máxima da poligonal de concreto perpendicular
* à direção da linha neutra (alfa+90 graus);
*e_CG: Saída da subrotina: Deformação do CG que possibilita que a força normal
* resistente seja igual à força normal solicitante; e_CG é expresso em por mil
* e convencionado positivo se for encurtamento. Admite-se que o CG esteja
* localizado na origem do sistema de coordenadas: é em relação a este ponto que
* são calculados os esforços resistentes da seção;
*Nr, Mrx, Mry: Saídas da subrotina: Esforços resistentes da seção (sist. Oxy):
* Nr >0 --> força normal de compressão;
* Mrx>0 --> momento definido por uma excentricidade positiva sobre o eixo x;
* Mry>0 --> momento definido por uma excentricidade positiva sobre o eixo y;
*EXISTE: Variável que assume 'SIM' ou 'NAO', conforme exista ou não um alfa que
* possibilite beta_r ser igual ao beta_s dado. Explica-se: Se se arbitrar uma
* n_solic alta e a seção for não simétrica em relação aos eixos x e y, pode
* ocorrer que o ângulo beta_r se restrinja à uma pequena faixa de variação
* mesmo que se arbitre alfa ao longo de todos os 360 graus. Logo, se beta_s não
* estiver dentro desta faixa, não haverá alfa que iguale beta_r ao beta_s dado.
* Para exemplificar, suponha uma seção assimétrica submetida à uma n_solic
* exatamente igual à máxima que a seção pode suportar: só há uma direção para o
* momento resistente (beta_r), qualquer que seja a direção arbitrada para a
* linha neutra; logo, para este caso, não se pode impor um beta_s (direção do
* momento solicitante) qualquer, pois não há solução matemática. Se a seção for
* simétrica em relação aos eixos x e y, não há problema: variando-se alfa de
* 360 graus, a faixa de variação de beta_r também será de 360 graus, e sempre
* haverá solução;
*-----
* SISTEMA FLEXOR - DIMENSIONAMENTO E VERIFICAÇÃO DE PILARES ESBELTOS DE
* CONCRETO ARMADO COM SEÇÃO QUALQUER SOLICITADOS POR FLEXÃO
* COMPOSTA OBLÍQUA
* DIAG. TENSÃOxDEFORM.DO CONCRETO: POLINÔMIOS DE GRAU QUALQUER
* CONSIDERAÇÃO DA FLUÊNCIA: TEORIA LINEAR DE FLUÊNCIA
* AÇOS BRASILEIROS DO TIPO 'A' OU 'B' COM QUALQUER fy OU Es
* Copyright(c) 1997, Ismael Wilson Cadamuro Júnior
*-----
SUBROUTINE PROCURA_ALFA (TETA,BETA_S,N_SOLIC,SIMETRIA,
+ N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,
+ X_BARRAS_ACO,Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,
+ MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,
+ ALFA,H_ALFA,e_CG,Nr,Mrx,Mry,EXISTE)
CHARACTER*1 CLASSE_ACO
CHARACTER*3 SIMETRIA
CHARACTER*3 EXISTE
INTEGER N_POLIG_CONCR,N_BARRAS_ACO,N_m,VETOR_m(N_m+1),N_CALL,
+ N_ITERACOES_MAX,SINAL,I,J,K
INTEGER*2 HOR,MIN,SEG,DSEG
REAL*8 TETA,BETA_S,N_SOLIC,X_POLIG_CONCR(N_POLIG_CONCR+1),
+ Y_POLIG_CONCR(N_POLIG_CONCR+1),X_BARRAS_ACO(N_BARRAS_ACO+1),
+ Y_BARRAS_ACO(N_BARRAS_ACO+1),A_BARRAS_ACO(N_BARRAS_ACO+1),
+ MATRIZ_C(N_m+1,30),MATRIZ_e_LIMITES(N_m+1,2),fc,FI_FLUENCIA,fy,

```

```

+ Es,ALFA,H_ALFA,e_CG,Nr,Mrx,Mry,a,b,x,ya,yb,yx,a0,ya0,M,
+ PRECISAO_x,PRECISAO_y,BETA_R,PI,PASSO,AUX1,AUX2,y_MIN,TIME1,
+ TIME2,DELTA_TIME
*
* Obs.: a dimensão dos vetores contém o "+1" para não ocorrer erro no caso
* de se entrar com n_polig_concr ou n_barras_aco igual a zero!
CALL GETTIM(HOR,MIN,SEG,DSEG) !comandos para cálc. do tempo
TIME1=HOR*60*60+MIN*60+SEG+DSEG/100. ! de execução da subrotina
EXISTE='SIM'
PI=ATAN2(0.,-1.)
PRECISAO_x=0.0001 !precisão de 0.0001 rad (0.0057 grau)
PRECISAO_y=0.0001 !precisão de 0.0001 rad (0.0057 grau)
N_ITERACOES_MAX=10000 !número máximo de iterações
N_CALL=0
IF (BETA_S.LT.-PI) BETA_S=BETA_S+2*PI
IF (BETA_S.GT.+PI) BETA_S=BETA_S-2*PI
*rotina especial para teta=0:
IF (ABS(TETA).LE.1E-10) THEN
IF (SIMETRIA.EQ.'SIM') THEN
BETA_R=BETA_S
Mrx=0
Mry=0
GOTO 1
ELSE
ALFA=0
N_CALL=N_CALL+1
CALL PROCURA_e_CG (TETA,ALFA,N_SOLIC,N_POLIG_CONCR,
+ N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,
+ Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,
+ MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,H_ALFA,e_CG
+ ,Nr,Mrx,Mry)
IF (ABS(Mrx).LE.1E-10.AND.ABS(Mry).LE.1E-10) THEN
BETA_R=BETA_S
Mrx=0
Mry=0
GOTO 1
END IF
BETA_R=ATAN2(Mry,Mrx)
yx=BETA_R-BETA_S
IF (yx.LT.-PI) yx=yx+2*PI
IF (yx.GT.+PI) yx=yx-2*PI
IF (ABS(yx).LE.PRECISAO_y) THEN
GOTO 1
ELSE
EXISTE='NAO'
GOTO 1
END IF
END IF
END IF
*cálc. do 'a' inicial:
a=BETA_S-PI/2 !cálc. do 'a' inicial: direção perpendicular
IF (a.LT.-PI) a=a+2*PI ! à do momento solicitante
IF (a.GT.+PI) a=a-2*PI
ALFA=a
N_CALL=N_CALL+1
CALL PROCURA_e_CG (TETA,ALFA,N_SOLIC,N_POLIG_CONCR,N_BARRAS_ACO,
+ X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,Y_BARRAS_ACO,
+ A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,fc,
+ FI_FLUENCIA,fy,Es,CLASSE_ACO,H_ALFA,e_CG,Nr,Mrx,Mry)
BETA_R=ATAN2(Mry,Mrx)
ya=BETA_R-BETA_S !cálc. 'ya' inicial
IF (ya.LT.-PI) ya=ya+2*PI
IF (ya.GT.+PI) ya=ya-2*PI
IF (ABS(ya).LE.PRECISAO_y) GOTO 1 !testa se 'ya' já não atende à precisão
*define se o alfa deve aumentar ou diminuir para poder igualar beta_r ao beta_s:
IF (SIN(BETA_R-BETA_S).LT.0) THEN
SINAL=+1
ELSE
SINAL=-1
END IF
IF (TETA.LT.0) SINAL=-SINAL
*cálc. do 'b' inicial: Método dos Intervalos Encaixantes
IF (SIMETRIA.EQ.'SIM') THEN
PASSO=SINAL*PI/12 !passo=15 graus na direção oposta à de beta_s
ELSE

```

```

        PASSO=SINAL*PI/36          !passo=5 graus na direção oposta à de beta_s
    END IF
    b=a
    yb=ya
    a0=a
    ya0=ya
    J=0
    DO WHILE (ya*yb.GT.0)          !cálc. do 'b' inicial: Método dos
        a=b                        ! Intervalos Encaixantes
        ya=yb
        b=b+PASSO
        IF (ABS(b-a0).GT.0.999*PI) THEN !alfa já foi incrementado em 180 graus e
            EXISTE='NAO'            ! não encontrou raiz --> não há raiz
            GOTO 1
        END IF
        ALFA=b
        N_CALL=N_CALL+1
        CALL PROCURA_e_CG (TETA,ALFA,N_SOLIC,N_POLIG_CONCR,N_BARRAS_ACO,
+       X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,Y_BARRAS_ACO,
+       A_BARRAS_ACO,N_m,VECTOR_m,MATRIZ_C,MATRIZ_e_LIMITES,fc,
+       FI_FLUENCIA,fy,Es,CLASSE_ACO,H_ALFA,e_CG,Nr,Mrx,Mrx)
        BETA_R=ATAN2(Mry,Mrx)
        yb=BETA_R-BETA_S          !cálc. 'yb' inicial
        IF (yb.LT.-PI) yb=yb+2*PI
        IF (yb.GT.+PI) yb=yb-2*PI
        IF (ya*yb.LT.0) THEN !testa se entre 'a' e 'b' há realmente uma raiz: yb
            x=(a+b)/2            ! pode mudar de sinal se, com o incremento de alfa,
            ALFA=x                ! beta_r for se afastando de beta_s (+ de 180graus)
            N_CALL=N_CALL+1
            CALL PROCURA_e_CG (TETA,ALFA,N_SOLIC,N_POLIG_CONCR,
+       N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,
+       Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VECTOR_m,MATRIZ_C,
+       MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,H_ALFA,e_CG
+       ,Nr,Mrx,Mrx)
            BETA_R=ATAN2(Mry,Mrx)
            yx=BETA_R-BETA_S      !cálc. de um ponto entre 'ya' e 'yb' ('yx')
            IF (yx.LT.-PI) yx=yx+2*PI
            IF (yx.GT.+PI) yx=yx-2*PI
            AUX1=yx-ya
            IF (AUX1.LT.0) AUX1=AUX1+2*PI
            AUX2=yb-ya
            IF (AUX2.LT.0) AUX2=AUX2+2*PI
            IF (AUX1.LE.AUX2) THEN !cálc. de y_min: o ponto 'yx' está à esquerda
                y_MIN=ya        ! (sentido trigonométrico, ou anti-horário)
                ELSE            ! do ponto 'y_min'; logo, para que haja
                y_MIN=yb        ! realmente uma raiz entre 'a' e 'b', é
            END IF              ! necessário que o ponto 'y_min' seja o
            IF (y_MIN.LE.0) THEN ! ponto negativo; caso contrário, o ponto
                IF (yb*yx.GE.0) THEN ! 'ya' passou para o 'yb' pelo lado oposto
                    b=x          ! ao do beta_s (beta_s + 180 graus)
                    yb=yx
                    ELSE        ! <----- há raiz entre 'a' e 'b'
                    a=x
                    ya=yx
                END IF
            ELSE
                a=b            ! <--- não há raiz entre 'a' e 'b'
                ya=yb
            END IF
        END IF
        J=J+1
        IF (J.GT.N_ITERACOES_MAX) THEN !teste de consistência
            CALL BEEPQQ(1500,1000)
            CALL BEEPQQ(1500,1000)
            WRITE(*,*) 'ERRO (SUBROTINA PROCURA_ALFA): ULTRAPASSADO O ',
+       'NUMERO MAXIMO DE ITERACOES'
            STOP
        END IF
        IF (ya*yb.LT.0) THEN
            IF (SIN(ya0)*SIN(ya).LT.0) THEN !testa se a raiz encontrada não é a
                EXISTE='NAO'            !raiz do ponto 'espelho' teta=-teta
                GOTO 1
            END IF
        END IF
    END IF

```

```

END DO
*ordena os valores de 'a' e 'b':
  IF (a.GT.b) THEN                                !ordenando os valores de 'a' e
    AUX1=a                                        ! de 'b': 'a' deve ser menor
    AUX2=b                                        ! que 'b'
    a=AUX2
    b=AUX1
    AUX1=ya
    AUX2=yb
    ya=AUX2
    yb=AUX1
  END IF
*processo iterativo de cálculo da raiz (Método da Posição Falsa):
  I=0
  K=0                                             !contador do número de iterações
  M=ya
  x=a                                             !x fictício para início do loop
  yx=ya                                           !yx fictício para início do loop
  AUX1=1
  DO WHILE (ABS(yx).GT.PRECISAO_y.OR.K.EQ.0)
    IF (M*yx.GE.0) THEN
      a=x
      ya=yx
    ELSE
      b=x
      yb=yx
    END IF
    IF (b-a.LE.PRECISAO_x) THEN                  !testa se b-a atende à precisão
      N_CALL=N_CALL+1                           !a raiz é a média dos dois valores
      K=K+1
      I=I+1
      x=(a+b)/2
      ALFA=x
      CALL PROCURA_e_CG (TETA,ALFA,N_SOLIC,N_POLIG_CONCR,
+       N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,
+       Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,
+       MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,H_ALFA,e_CG
+       ,Nr,Mrx,Mry)
      BETA_R=ATAN2(Mry,Mrx)
      GOTO 1
    END IF
    x=(a*yb-b*ya)/(yb-ya)                       !cálc. de 'x'
    IF (I.EQ.8) THEN
      x=(a+b)/2
      I=0
    END IF
    N_CALL=N_CALL+1
    K=K+1
    I=I+1
    ALFA=x
    CALL PROCURA_e_CG (TETA,ALFA,N_SOLIC,N_POLIG_CONCR,N_BARRAS_ACO,
+     X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,Y_BARRAS_ACO,
+     A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,fc,
+     FI_FLUENCIA,fy,Es,CLASSE_ACO,H_ALFA,e_CG,Nr,Mrx,Mry)
    BETA_R=ATAN2(Mry,Mrx)
    yx=BETA_R-BETA_S                             !cálc. de 'yx'
    IF (yx.LT.-PI) yx=yx+2*PI
    IF (yx.GT.+PI) yx=yx-2*PI
    IF (K.GT.N_ITERACOES_MAX) THEN               !teste de consistência
      CALL BEEPQQ(1500,1000)
      CALL BEEPQQ(1500,1000)
      WRITE(*,*) 'ERRO (SUBROTINA PROCURA_ALFA): ULTRAPASSADO O ',
+ ' NUMERO MAXIMO DE ITERACOES'
      STOP
    END IF
  END DO
*cálculos e testes finais da subrotina:
  1 CALL GETTIM(HOR,MIN,SEG,DSEG)                !comandos para cálc. do tempo
  TIME2=HOR*60*60+MIN*60+SEG+DSEG/100.         ! de execução da subrotina
  IF (TIME2.LT.TIME1) TIME2=TIME2+24*60*60
  DELTA_TIME=TIME2-TIME1
***** WRITE(*,2) K,N_CALL,DELTA_TIME
  2 FORMAT(' PROCURA_ALFA: ',I4,' iteracoes ',I4,
+ ' chamadas ',F8.2,' segundos')              !mensagem informativa

```

```

IF (ALFA.LT.-PI) ALFA=ALFA+2*PI
IF (ALFA.GT.+PI) ALFA=ALFA-2*PI
yx=BETA_R-BETA_S
IF (yx.LT.-PI) yx=yx+2*PI
IF (yx.GT.+PI) yx=yx-2*PI
IF (ABS(yx).GT.4*PRECISAO_Y.AND.EXISTE.EQ.'SIM') THEN
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) '-----'
WRITE(*,*) 'AVISO (SUBROTINA PROCURA_ALFA): A SUBROTINA ACHOU ',
+ 'UMA RAIZ EM QUE BETA_R EH DIFERENTE DE BETA_S.'
WRITE(*,*) 'HIPOTHESES:'
WRITE(*,*) ' -PASSO DE 5 GRAUS EH MUITO GRANDE'
WRITE(*,*) ' -PRECISAO_x FOI DETERMINANTE: (b-a) < PRECISAO_x'
WRITE(*,*) ' -IMPRECISAO NUMERICA: LN QUASE PARALELA C/ LADO'
WRITE(*,*) ' -IMPRECISAO NUMERICA: TETA MUITO PEQUENO'
WRITE(*,*) '-----'
WRITE(*,*) 'BETA_R = ',ATAN2(Mry,Mrx)*180/PI,' graus'
WRITE(*,*) 'BETA_S = ',BETA_S*180/PI,' graus'
WRITE(*,*) '-----'
WRITE(*,*) 'a=',a*180/PI,' grs. -> BETA_R_a=',(ya+BETA_S)*180/PI
WRITE(*,*) 'b=',b*180/PI,' grs. -> BETA_R_b=',(yb+BETA_S)*180/PI
WRITE(*,*) 'ALFA = ',ALFA*180/PI,' graus'
WRITE(*,*) 'TETA = ',TETA
WRITE(*,*) 'e_CG = ',e_CG,' por mil'
WRITE(*,*) 'Mrx = ',Mrx
WRITE(*,*) 'Mry = ',Mry
WRITE(*,*) '-----'
WRITE(*,*) 'Entre com opcao:'
WRITE(*,*) '      1 --> continuar assim mesmo'
WRITE(*,*) '      2 --> admite que, para o teta arbitrado, ',
+ 'nao exista alfa que propicie igualar beta_r a beta_s'
READ(*,*) x
IF (x.EQ.2) EXISTE='NAO'
END IF
END
=====fim da subrotina PROCURA_ALFA=====

*=====SUBROTINA PROCURA_e_CG=====
*Subrotina que, dada a geometria da seção (sist. Oxy), a curvatura, a direção da
* linha neutra e a força normal solicitante, calcula o e_cg (deformação do CG
* da seção) que faz com que a força normal resistente da seção equilibre a
* força normal solicitante. Para este estado de deformação, calcula também os
* esforços resistentes da seção (sist. Oxy). Atenção: esta subrotina não
* obedece a nenhuma restrição de norma quanto aos limites para e_cg, podendo,
* portanto, o e_cg calculado ultrapassar o ELU de ruptura do concreto
* comprimido ou de deformação plástica excessiva da armadura tracionada.
* O cálculo é feito por tentativas: arbitra-se e_cg, calcula-se a força normal
* resistente (subrotina esforcos_secao) e compara-se com a força normal
* solicitante. Para otimizar este processo, esta subrotina utiliza um método
* numérico iterativo de cálculo de zeros de funções: o Método da Posição Falsa
* (ver Ruggiero, M.A.G. e Lopes, V.L.R. - "Cálculo Numérico - aspectos teóricos
* e computacionais" - pg.37). Este método foi escolhido por ser eficiente e por
* ter convergência garantida para o tipo de função em questão
* N_resistente=f(e_CG), ao contrário de outros métodos mais eficientes, como o
* de Newton-Raphson ou o da Secante. Para o entendimento da subrotina,
* recomenda-se consultar a bibliografia indicada acima.
*Unidades de entrada: Sistema de unidades qualquer, desde que consistente;
*TETA: Curvatura majorada adimensional (1000h_alfa/r) - direção perpendicular
* à da linha neutra; teta pode ser positivo, negativo ou zero;
*ALFA: Ângulo (em radianos) de rotação da linha neutra, medido a partir do eixo
* x no sentido anti-horário;
*N_SOLIC: Força normal que solicita a seção; positiva --> compressão;
* se As=0, n_solic não pode ser de tração;
*N_POLIG_CONCR e N_BARRAS_ACO: Número de nós da poligonal de concreto (incluindo
* o nó de chegada, que é o mesmo nó de saída) e número de barras de aço da
* seção;
*X_POLIG_CONCR, Y_POLIG_CONCR e X_BARRAS_ACO,Y_BARRAS_ACO: Vetores das
* coordenadas (sistema Oxy, ou seja, não rotacionado) dos nós da poligonal de
* concreto (a poligonal deve ser numerada no sentido anti-horário, com vazios
* no sentido horário; com o nó de chegada, que é o mesmo nó de saída) e das
* coordenadas das barras de aço;

```

```

*A_BARRAS_ACO: Vetor das áreas das barras de aço;
*N_m: Número de curvas (polinômios de grau qualquer) que compoem o diagrama
* tensão-deformação adotado para o concreto; cada curva é definida por um
* somatório do tipo: sigma_c=fc(C0.e^0 + C1.e^1 + C2.e^2 +...+Cm.e^m); para o
* diagrama parábola-retângulo da NBl/78, temos N_m=2 (curva parabólica e curva
* reta constante);
*VETOR_m: Vetor que contém os graus dos N_m polinômios;para o parábola-retângulo
* da NBl/78, temos VETOR_m=(2,0) (m=2 -> parábola; m=0 -> reta constante);
*MATRIZ_C: Matriz que contém os coeficientes dos N_m polinômios: linha i contém
* os Ck (k=1 à m+1) coeficientes do polinômio i; para o parábola-retângulo da
* NBl/78, temos: MATRIZ_C = !0 1 -0.25! -> coef. da parábola
* !1 0 0 ! -> coef. da reta constante;
*MATRIZ_e_LIMITES: Matriz que contém as deformações (em por mil) iniciais e
* finais onde age cada um dos N_m polinômios (deformações estas sem o efeito da
* fluência): linha i contém as deformações iniciais e finais onde age o
* polinômio i; OBS: colocar um número grande para a deformação final do último
* polinômio que compõe o diagrama tensão-deformação adotado para o concreto,
* não finalizando o diagrama no ELU; para o parábola-retângulo da NBl/78,temos:
* MATRIZ_e_LIMITES = !0 2 ! -> def. limites da parábola
* !2 1000! -> def. limites da reta constante;
*fc: Tensão máxima de compressão à que o concreto pode estar sujeito, ou seja,
* é a tensão máxima do diagrama tensão-deformação adotado para o concreto.
* Não há censo quanto ao valor a se utilizar. A NBl/78 indica fc=0.85fcd
* (ELU), mas, para o estudo da estabilidade, alguns autores indicam fc=0.85fck;
*FI_FLUENCIA: Função de Fluência (Teoria Linear de Fluência);
*fy: Tensão de escoamento do aço (valor de cálculo ou não);
*Es: Módulo de Elasticidade do aço (usualmente igual a 210000MPa);
*CLASSE_ACO: É a classe à que pertence o aço (A ou B);
*H_ALFA:Saída da subrotina: Altura máxima da poligonal de concreto perpendicular
* à direção da linha neutra (alfa+90 graus);
*e_CG: Saída da subrotina: Deformação do CG que possibilita que a força normal
* resistente seja igual à força normal solicitante. e_cg é expresso em por mil
* e convencionalmente positivo se for encurtamento. Admite-se que o CG esteja
* localizado na origem do sistema de coordenadas: é em relação a este ponto que
* são calculados os esforços resistentes da seção;
*Nr, Mrx, Mry: Saídas da subrotina: Esforços resistentes da seção (sist. Oxy):
* Nr >0 --> força normal de compressão;
* Mrx>0 --> momento definido por uma excentricidade positiva sobre o eixo x;
* Mry>0 --> momento definido por uma excentricidade positiva sobre o eixo y;
-----
* SISTEMA FLEXOR - DIMENSIONAMENTO E VERIFICAÇÃO DE PILARES ESBELTOS DE
* CONCRETO ARMADO COM SEÇÃO QUALQUER SOLICITADOS POR FLEXÃO
* COMPOSTA OBLÍQUA
* DIAG. TENSÃOxDEFORM.DO CONCRETO: POLINÔMIOS DE GRAU QUALQUER
* CONSIDERAÇÃO DA FLUÊNCIA: TEORIA LINEAR DE FLUÊNCIA
* AÇOS BRASILEIROS DO TIPO 'A' OU 'B' COM QUALQUER fy OU Es
* Copyright(c) 1997, Ismael Wilson Cadamuro Júnior
-----
SUBROUTINE PROCURA_e_CG (TETA,ALFA,N_SOLIC,N_POLIG_CONCR,
+ N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,
+ Y_BARRAS_ACO,A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,
+ fc,FI_FLUENCIA,fy,Es,CLASSE_ACO,H_ALFA,e_CG,Nr,Mrx,Mry)
CHARACTER*1 CLASSE_ACO
INTEGER N_POLIG_CONCR,N_BARRAS_ACO,N_m,VETOR_m(N_m+1),K,I,N_CALL,
+ N_ITERACOES_MAX
INTEGER*2 HOR,MIN,SEG,DSEG
REAL*8 X_POLIG_CONCR(N_POLIG_CONCR+1),Y_POLIG_CONCR(N_POLIG_CONCR+
+ 1),X_BARRAS_ACO(N_BARRAS_ACO+1),Y_BARRAS_ACO(N_BARRAS_ACO+1),
+ U_POLIG_CONCR[ALLOCATABLE](:),V_POLIG_CONCR[ALLOCATABLE](:),
+ U_BARRAS_ACO[ALLOCATABLE](:),V_BARRAS_ACO[ALLOCATABLE](:),
+ A_BARRAS_ACO(N_BARRAS_ACO+1),MATRIZ_C(N_m+1,30),
+ MATRIZ_e_LIMITES(N_m+1,2),fc,FI_FLUENCIA,fy,Es,e_CG,Nr,Mrx,Mry,
+ TETA,ALFA,N_SOLIC,Mrx,Mrv,e_CG_MINIMO,e_CG_MAXIMO,Nr_MINIMO,
+ Nr_MAXIMO,a,b,ya,yb,yx,M,PRECISAO_x,PRECISAO_y,x,PASSO,H_ALFA,
+ AUX1,AUX2,TIME1,TIME2,DELTA_TIME
ALLOCATE (U_POLIG_CONCR(2*N_POLIG_CONCR+1),
+ V_POLIG_CONCR(2*N_POLIG_CONCR+1),U_BARRAS_ACO(N_BARRAS_ACO+1),
+ V_BARRAS_ACO(N_BARRAS_ACO+1))
*
* Obs.: a dimensão dos vetores contém o "+1" para não ocorrer erro no caso
* de se entrar com n_polig_concr ou n_barras_aco igual a zero!
CALL GETTIM(HOR,MIN,SEG,DSEG) !comandos para cálc. do tempo
TIME1=HOR*60*60+MIN*60+SEG+DSEG/100. ! de execução da subrotina
N_ITERACOES_MAX=10000 !número máximo de iterações
PRECISAO_x=0.0000001 !precisão de 0.0000001 por mil

```

```

PRECISAO_y=0.0001 !precis.de 0.01% (geralmente, é a precisão que predomina
N_CALL=0
*rotação do sistema de coord. Oxy para o sistema Ouv: (LN paralela ao eixo u)
CALL ROTACAO (ALFA,N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,
+ Y_POLIG_CONCR,X_BARRAS_ACO,Y_BARRAS_ACO,U_POLIG_CONCR,
+ V_POLIG_CONCR,U_BARRAS_ACO,V_BARRAS_ACO)
CALL MAXIMO_MINIMO_DIFEREN(N_POLIG_CONCR,V_POLIG_CONCR,a,b,H_ALFA)
*cálc. de 'a' e 'b' iniciais: (e_CG limites estimados inicialmente com base no
*ELU de ruptura)
CALL TETA_eCG_ELU (TETA,N_POLIG_CONCR,N_BARRAS_ACO, !e_CG limites
+ V_POLIG_CONCR,V_BARRAS_ACO,FI_FLUENCIA,e_CG_MINIMO,e_CG_MAXIMO)
IF (e_CG_MINIMO.LT.-10.OR.e_CG_MINIMO.GT.+10) e_CG_MINIMO=-10
IF (e_CG_MAXIMO.LT.-10.OR.e_CG_MAXIMO.GT.+10) e_CG_MAXIMO=+10
AUX1=e_CG_MINIMO
AUX2=e_CG_MAXIMO
IF (e_CG_MINIMO.GT.e_CG_MAXIMO) THEN
e_CG_MINIMO=AUX2
e_CG_MAXIMO=AUX1
END IF
IF (e_CG_MAXIMO.LT.0) e_CG_MAXIMO=-e_CG_MAXIMO
N_CALL=N_CALL+1 !cálc. Nr_mínimo
CALL ESFORCOS_SECAO (N_POLIG_CONCR,N_BARRAS_ACO,U_POLIG_CONCR,
+ V_POLIG_CONCR,U_BARRAS_ACO,V_BARRAS_ACO,A_BARRAS_ACO,N_m,
+ VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,
+ CLASSE_ACO,e_CG_MINIMO,TETA,H_ALFA,Nr_MINIMO,Mru,Mrv)
N_CALL=N_CALL+1 !cálc. Nr_máximo
CALL ESFORCOS_SECAO (N_POLIG_CONCR,N_BARRAS_ACO,U_POLIG_CONCR,
+ V_POLIG_CONCR,U_BARRAS_ACO,V_BARRAS_ACO,A_BARRAS_ACO,N_m,
+ VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,
+ CLASSE_ACO,e_CG_MAXIMO,TETA,H_ALFA,Nr_MAXIMO,Mru,Mrv)
PASSO=0.1 !caso o e_CG_MAXIMO estimado pelo ELU não
I=1 ! permita o equilíbrio:
DO WHILE (N_SOLIC.GT.Nr_MAXIMO) ! ... Método dos Intervalos Encaixantes
e_CG_MINIMO=e_CG_MAXIMO
Nr_MINIMO=Nr_MAXIMO
N_CALL=N_CALL+1
e_CG_MAXIMO=e_CG_MAXIMO+PASSO
CALL ESFORCOS_SECAO (N_POLIG_CONCR,N_BARRAS_ACO,U_POLIG_CONCR,
+ V_POLIG_CONCR,U_BARRAS_ACO,V_BARRAS_ACO,A_BARRAS_ACO,N_m,
+ VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,
+ CLASSE_ACO,e_CG_MAXIMO,TETA,H_ALFA,Nr_MAXIMO,Mru,Mrv) !cálc Nr_máximo
I=I+1
IF (I.EQ.100) PASSO=1
IF (I.GT.10000) EXIT
END DO
IF (N_SOLIC.GT.Nr_MAXIMO) THEN !teste de consistência
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) '-----'
WRITE(*,*) 'AVISO (SUBROTINA PROCURA_e_CG): IMPOSSIVEL EQUILI',
+ 'BRAR A FORCA NORMAL SOLICITANTE USANDO A CURVATURA ARBITRADA'
WRITE(*,*) 'TETA ARBITRADO =',TETA
WRITE(*,*) 'Se quiser continuar assim mesmo, tecle <ENTER>.'
WRITE(*,*) '-----'
PAUSE
GOTO 1
END IF
PASSO=0.1 !caso o e_CG_MINIMO estimado pelo ELU não
I=1 ! permita o equilíbrio:
DO WHILE (N_SOLIC.LT.Nr_MINIMO) ! ... Método dos Intervalos Encaixantes
e_CG_MAXIMO=e_CG_MINIMO
Nr_MAXIMO=Nr_MINIMO
N_CALL=N_CALL+1
e_CG_MINIMO=e_CG_MINIMO-PASSO
CALL ESFORCOS_SECAO (N_POLIG_CONCR,N_BARRAS_ACO,U_POLIG_CONCR,
+ V_POLIG_CONCR,U_BARRAS_ACO,V_BARRAS_ACO,A_BARRAS_ACO,N_m,
+ VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,
+ CLASSE_ACO,e_CG_MINIMO,TETA,H_ALFA,Nr_MINIMO,Mru,Mrv) !cálc Nr_mínimo
I=I+1
IF (I.EQ.100) PASSO=1
IF (I.GT.10000) EXIT
END DO
IF (N_SOLIC.LT.Nr_MINIMO) THEN !teste de consistência
CALL BEEPQQ(1500,1000)

```



```

CALL ESFORCOS_SECAO (N_POLIG_CONCR,N_BARRAS_ACO,U_POLIG_CONCR,
+   V_POLIG_CONCR,U_BARRAS_ACO,V_BARRAS_ACO,A_BARRAS_ACO,N_m,
+   VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,fc,FI_FLUENCIA,fy,Es,
+   CLASSE_ACO,e_CG,TETA,H_ALFA,Nr,Mru,Mrv)
GOTO 1
END IF
IF (K.GT.N_ITERACOES_MAX) THEN                                !teste de consistência
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) 'ERRO (SUBROTINA PROCURA_e_CG): ULTRAPASSADO O ',
+ 'NUMERO MAXIMO DE ITERACOES'
STOP
END IF
END DO
*cálculos e testes finais da subrotina:
1 CALL GETTIM(HOR,MIN,SEG,DSEG)                                !comandos para cálc. do tempo
TIME2=HOR*60*60+MIN*60+SEG+DSEG/100.                          ! de execução da subrotina
IF (TIME2.LT.TIME1) TIME2=TIME2+24*60*60
DELTA_TIME=TIME2-TIME1
***** WRITE(*,2) K,N_CALL,DELTA_TIME
!
2 FORMAT('          PROCURA_e_CG:          ',I4,' iteracoes          ',I4,
+ ' chamadas          ',F8.2,' segundos')                      !mensagem informativa
Mrx=Mru*COS(ALFA)-Mrv*SIN(ALFA)                                !cálc. dos esforços no sistema Oxy
Mry=Mru*SIN(ALFA)+Mrv*COS(ALFA)
IF (ABS(Nr-N_SOLIC).GT.ABS(1.05*PRECISAO_y*N_SOLIC+1E-10).AND.
+ ABS(b-a).GT.PRECISAO_x+1E-10) THEN
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) '-----'
WRITE(*,*) 'AVISO (SUBROTINA PROCURA_e_CG): A SUBROTINA ',
+ 'ACHOU UMA RAIZ EM QUE Nr EH DIFERENTE DE N_SOLIC.'
WRITE(*,*) 'HIPOTESIS:'
WRITE(*,*) ' -PRECISAO_x PREDOMINOU SOBRE A PRECISAO_y'
WRITE(*,*) '-----'
WRITE(*,*) 'Nr          = ',Nr
WRITE(*,*) 'N_SOLIC= ',N_SOLIC
WRITE(*,*) '-----'
WRITE(*,*) 'Mrx          = ',Mrx
WRITE(*,*) 'Mry          = ',Mry
WRITE(*,*) 'TETA          = ',TETA
WRITE(*,*) 'ALFA          = ',ALFA*180/ATAN2(0.,-1.),' graus'
WRITE(*,*) 'H_ALFA          = ',H_ALFA
WRITE(*,*) 'e_CG          = ',e_CG,' por mil'
WRITE(*,*) '-----'
WRITE(*,*) 'Entre com opcao:'
WRITE(*,*) '          1 --> continuar assim mesmo'
WRITE(*,*) '          --> aperte control+break para parar'
READ(*,*) I
END IF
DEALLOCATE (U_POLIG_CONCR,V_POLIG_CONCR,U_BARRAS_ACO,V_BARRAS_ACO)
END                                                                !desalocando a memória
=====fim da subrotina PROCURA_e_CG=====

```

```

=====SUBROTINA TRANSLACAO=====
*Subrotina que faz uma translação no sistema de coordenadas: dada as coordenadas
* dos nós da poligonal de concreto e das barras de aço no sistema Oxy, calcula
* essas coordenadas no sistema Oxyt (transladado de x_transl e y_transl em
* relação ao sistema Oxy).
*X_TRANSL, Y_TRANSL: Coordenadas da origem do novo sistema de coordenadas;
*N_POLIG_CONCR e N_BARRAS_ACO: Número de nós da poligonal de concreto (incluindo
* o nó de chegada, que é o mesmo nó de saída) e número de barras de aço da
* seção;
*X_POLIG_CONCR, Y_POLIG_CONCR e X_BARRAS_ACO,Y_BARRAS_ACO: Vetores das
* coordenadas (sistema Oxy) dos nós da seção de concreto e das barras de aço;
*Xt_POLIG_CONCR, Yt_POLIG_CONCR e Xt_BARRAS_ACO,Yt_BARRAS_ACO: Saídas da
* subrotina: Vetores das coordenadas (sistema Oxyt, ou seja, transladado em
* relação ao sistema Oxy) dos nós da seção de concreto e das das barras de aço;
*-----
* SISTEMA FLEXOR - DIMENSIONAMENTO E VERIFICAÇÃO DE PILARES ESBELTOS DE
* CONCRETO ARMADO COM SEÇÃO QUALQUER SOLICITADOS POR FLEXÃO
* COMPOSTA OBLÍQUA

```

```

*           DIAG. TENSÃOxDEFORM.DO CONCRETO: POLINÔMIOS DE GRAU QUALQUER
*           CONSIDERAÇÃO DA FLUÊNCIA: TEORIA LINEAR DE FLUÊNCIA
*           AÇOS BRASILEIROS DO TIPO 'A' OU 'B' COM QUALQUER fy OU Es
* Copyright(c) 1997, Ismael Wilson Cadamuro Júnior

```

```

-----
SUBROUTINE TRANSLACAO (X_TRANSL,Y_TRANSL,N_POLIG_CONCR,
+ N_BARRAS_ACO,X_POLIG_CONCR,Y_POLIG_CONCR,X_BARRAS_ACO,
+ Y_BARRAS_ACO,Xt_POLIG_CONCR,Yt_POLIG_CONCR,Xt_BARRAS_ACO,
+ Yt_BARRAS_ACO)
INTEGER N_POLIG_CONCR,N_BARRAS_ACO,I
REAL*8 X_POLIG_CONCR(N_POLIG_CONCR+1),Y_POLIG_CONCR(N_POLIG_CONCR+
+ 1),X_BARRAS_ACO(N_BARRAS_ACO+1),Y_BARRAS_ACO(N_BARRAS_ACO+1),
+ Xt_POLIG_CONCR(N_POLIG_CONCR+1),Yt_POLIG_CONCR(N_POLIG_CONCR+1),
+ Xt_BARRAS_ACO(N_BARRAS_ACO+1),Yt_BARRAS_ACO(N_BARRAS_ACO+1),
+ X_TRANSL,Y_TRANSL
* Obs.: a dimensão dos vetores contém o "+1" para não ocorrer erro no caso
* de se entrar com n_polig_concr ou n_barras_aco igual a zero!
I=1
DO WHILE (I.LE.N_POLIG_CONCR) !transl. das coord. da poligonal de concreto
Xt_POLIG_CONCR(I)=X_POLIG_CONCR(I)-X_TRANSL
Yt_POLIG_CONCR(I)=Y_POLIG_CONCR(I)-Y_TRANSL
I=I+1
END DO
I=1
DO WHILE (I.LE.N_BARRAS_ACO) !translação das coord. das barras de aço
Xt_BARRAS_ACO(I)=X_BARRAS_ACO(I)-X_TRANSL
Yt_BARRAS_ACO(I)=Y_BARRAS_ACO(I)-Y_TRANSL
I=I+1
END DO
END

```

=====fim da subrotina TRANSLACAO=====

```

=====SUBROTINA ROTACAO=====
*Subrotina que faz uma rotação no sistema de coordenadas: dada as coordenadas
* dos nós da poligonal de concreto e das barras de aço no sistema Oxy, calcula
* essas coordenadas no sistema Ouv (rotacionado de um ângulo alfa no sentido
* trigonométrico (anti-horário) em relação ao sistema Oxy).
*ALFA: Ângulo de rotação (em radianos) medido no sentido anti-horário (do eixo x
* para o eixo u);
*N_POLIG_CONCR e N_BARRAS_ACO: Número de nós da poligonal de concreto (incluindo
* o nó de chegada, que é o mesmo nó de saída) e número de barras de aço da
* seção;
*X_POLIG_CONCR, Y_POLIG_CONCR e X_BARRAS_ACO,Y_BARRAS_ACO: Vetores das
* coordenadas (sistema Oxy) dos nós da seção de concreto e das barras de aço;
*U_POLIG_CONCR, V_POLIG_CONCR e U_BARRAS_ACO,V_BARRAS_ACO: Saídas da subrotina:
* Vetores das coordenadas (sistema Ouv, ou seja, rotacionado em relação ao
* sistema Oxy) dos nós da seção de concreto e das das barras de aço;

```

```

-----
* SISTEMA FLEXOR - DIMENSIONAMENTO E VERIFICAÇÃO DE PILARES ESBELTOS DE
* CONCRETO ARMADO COM SEÇÃO QUALQUER SOLICITADOS POR FLEXÃO
* COMPOSTA OBLÍQUA
* DIAG. TENSÃOxDEFORM.DO CONCRETO: POLINÔMIOS DE GRAU QUALQUER
* CONSIDERAÇÃO DA FLUÊNCIA: TEORIA LINEAR DE FLUÊNCIA
* AÇOS BRASILEIROS DO TIPO 'A' OU 'B' COM QUALQUER fy OU Es
* Copyright(c) 1997, Ismael Wilson Cadamuro Júnior

```

```

-----
SUBROUTINE ROTACAO (ALFA,N_POLIG_CONCR,N_BARRAS_ACO,X_POLIG_CONCR,
+ Y_POLIG_CONCR,X_BARRAS_ACO,Y_BARRAS_ACO,U_POLIG_CONCR,
+ V_POLIG_CONCR,U_BARRAS_ACO,V_BARRAS_ACO)
INTEGER N_POLIG_CONCR,N_BARRAS_ACO,I
REAL*8 X_POLIG_CONCR(N_POLIG_CONCR+1),Y_POLIG_CONCR(N_POLIG_CONCR+
+ 1),X_BARRAS_ACO(N_BARRAS_ACO+1),Y_BARRAS_ACO(N_BARRAS_ACO+1),
+ U_POLIG_CONCR(N_POLIG_CONCR+1),V_POLIG_CONCR(N_POLIG_CONCR+1),
+ U_BARRAS_ACO(N_BARRAS_ACO+1),V_BARRAS_ACO(N_BARRAS_ACO+1),ALFA
* Obs.: a dimensão dos vetores contém o "+1" para não ocorrer erro no caso
* de se entrar com n_polig_concr ou n_barras_aco igual a zero!
I=1
DO WHILE (I.LE.N_POLIG_CONCR) !rotação das coord. da poligonal de concreto
U_POLIG_CONCR(I)=X_POLIG_CONCR(I)*COS(ALFA)+
+ Y_POLIG_CONCR(I)*SIN(ALFA)
V_POLIG_CONCR(I)=Y_POLIG_CONCR(I)*COS(ALFA)-
+ X_POLIG_CONCR(I)*SIN(ALFA)

```

```

        I=I+1
    END DO
    I=1
    DO WHILE (I.LE.N_BARRAS_ACO)          !rotação das coord. das barras de aço
        U_BARRAS_ACO(I)=X_BARRAS_ACO(I)*COS(ALFA)+
    +   Y_BARRAS_ACO(I)*SIN(ALFA)
        V_BARRAS_ACO(I)=Y_BARRAS_ACO(I)*COS(ALFA)-
    +   X_BARRAS_ACO(I)*SIN(ALFA)
        I=I+1
    END DO
    END
=====fim da subrotina ROTACAO=====

```

```

=====SUBROTINA TETA_eCG_ELU=====
*Subrotina que, dada a curvatura à qual a seção está sujeita, calcula os limites
* mínimo e máximo de e_cg (deformação em por mil do ponto de coordenadas (0,0),
* onde se admite estar o centro de gravidade da seção) para que a seção atinja
* o ELU de ruptura do concreto comprimido ou de deformação plástica excessiva
* da armadura tracionada. Admite-se que a linha neutra seja paralela ao eixo das
* abscissas (u).
*ATENÇÃO: Esta subrotina foi confeccionada com base nos domínios de deformação
* definidos na norma NBl/78;
*Unidades de entrada: Sistema de unidades qualquer, desde que consistente;
*TETA: Curvatura majorada adimensional (1000h/r); teta pode ser positivo,
* negativo ou zero;
*N_POLIG_CONCR e N_BARRAS_ACO: Número de nós da poligonal de concreto (incluindo
* o nó de chegada, que é o mesmo nó de saída) e número de barras de aço da
* seção;
*V_POLIG_CONCR e V_BARRAS_ACO: Vetores das ordenadas dos nós da seção de
* concreto e das barras de aço;
*FI_FLUENCIA: Função de Fluência (Teoria Linear de Fluência);
*e_CG_MINIMO e e_CG_MAXIMO: Saídas da subrotina: Limites mínimo e máximo de e_cg
* para que a seção não seja considerada rompida (Estado Limite Último de
* ruptura do concreto comprimido ou de deformação plástica excessiva da
* armadura tracionada); deformação de encurtamento --> positiva
-----
* SISTEMA FLEXOR - DIMENSIONAMENTO E VERIFICAÇÃO DE PILARES ESBELTOS DE
* CONCRETO ARMADO COM SEÇÃO QUALQUER SOLICITADOS POR FLEXÃO
* COMPOSTA OBLÍQUA
* DIAG. TENSÃOxDEFORM.DO CONCRETO: POLINÔMIOS DE GRAU QUALQUER
* CONSIDERAÇÃO DA FLUÊNCIA: TEORIA LINEAR DE FLUÊNCIA
* AÇOS BRASILEIROS DO TIPO 'A' OU 'B' COM QUALQUER fy OU Es
* Copyright(c) 1997, Ismael Wilson Cadamuro Júnior
-----

```

```

SUBROUTINE TETA_eCG_ELU (TETA,N_POLIG_CONCR,N_BARRAS_ACO,
+ V_POLIG_CONCR,V_BARRAS_ACO,FI_FLUENCIA,e_CG_MINIMO,e_CG_MAXIMO)
INTEGER N_POLIG_CONCR,N_BARRAS_ACO
REAL*8 V_POLIG_CONCR(N_POLIG_CONCR+1),V_BARRAS_ACO(N_BARRAS_ACO+1)
+ ,FI_FLUENCIA,e_CG_MINIMO,e_CG_MAXIMO,TETA,H_ALFA,V_CONCR_MAX,
+ V_CONCR_MIN,V_BARRAS_MAX,V_BARRAS_MIN,AUX1,AUX2
* Obs.: a dimensão dos vetores contém o "+1" para não ocorrer erro no caso
* de se entrar com n_polig_concr ou n_barras_aco igual a zero!
IF (N_POLIG_CONCR.EQ.0.AND.N_BARRAS_ACO.EQ.0) THEN !teste de consistência
    e_CG_MINIMO=0
    e_CG_MAXIMO=0
    GOTO 1
END IF
IF (N_POLIG_CONCR.LE.3.AND.N_POLIG_CONCR.NE.0) THEN !teste de consistência
    CALL BEEPQQ(1500,1000)
    CALL BEEPQQ(1500,1000)
    WRITE(*,*) 'ERRO (SUBROTINA TETA_eCG_ELU): POLIGONAL COM',
+ 'MENOS DE 4 NOS'
    STOP
END IF
IF (TETA.GE.0) THEN !cálc. h_alfa, v_concr_max e v_barras_min
    CALL MAXIMO_MINIMO_DIFEREN (N_POLIG_CONCR,V_POLIG_CONCR,
+ V_CONCR_MAX,V_CONCR_MIN,H_ALFA)
    CALL MAXIMO_MINIMO_DIFEREN (N_BARRAS_ACO,V_BARRAS_ACO,
+ V_BARRAS_MAX,V_BARRAS_MIN,AUX2)
ELSE !se teta<0 -> cálculo com a seção invertida
    CALL MAXIMO_MINIMO_DIFEREN (N_POLIG_CONCR,V_POLIG_CONCR,
+ V_CONCR_MIN,V_CONCR_MAX,H_ALFA)

```

```

CALL MAXIMO_MINIMO_DIFEREN (N_BARRAS_ACO,V_BARRAS_ACO,
+ V_BARRAS_MIN,V_BARRAS_MAX,AUX2)
END IF
IF (N_BARRAS_ACO.EQ.0) V_BARRAS_MIN=V_CONCR_MIN !não há barras de aço
IF (N_POLIG_CONCR.EQ.0) THEN !não há poligonal de concreto
V_CONCR_MAX=V_BARRAS_MAX
H_ALFA=AUX2
END IF
IF (H_ALFA.EQ.0) THEN !teste de consistência
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) 'ERRO (SUBROTINA TETA_eCG_ELU): ALTURA DA SECAO',
+ 'IGUAL A ZERO'
STOP
END IF
e.CG_MINIMO=-TETA/H_ALFA*V_BARRAS_MIN-10 !cálc. de e_cg_mínimo: deform. 10
AUX1=-TETA/H_ALFA*V_CONCR_MAX+3.5*(1+FI_FLUENCIA) !por mil no aço comanda
IF (TETA.GE.0) THEN !cálc. de e_cg_máximo:
AUX2=-TETA/H_ALFA*(V_CONCR_MAX-3./7*H_ALFA)+2*(1+FI_FLUENCIA)
ELSE !se teta<0 -> cálculo com a seção invertida
AUX2=-TETA/H_ALFA*(V_CONCR_MAX+3./7*H_ALFA)+2*(1+FI_FLUENCIA)
END IF
IF (AUX1.LE.AUX2) THEN
e.CG_MAXIMO=AUX1 !def. 3.5 por mil comanda
ELSE
e.CG_MAXIMO=AUX2 !def. 2 por mil comanda
END IF
IF (N_POLIG_CONCR.EQ.0) e.CG_MAXIMO=-TETA/H_ALFA*V_BARRAS_MAX+10
1 END
=====fim da subrotina TETA_eCG_ELU=====

```

```

=====SUBROTINA ESFORCOS_SECAO=====
*Subrotina que calcula os esforços resistentes totais (concreto e aço) para uma
* seção com geometria e estado de deformação dados.Admite-se que a linha neutra
* seja paralela ao eixo das abssissas (u). Esta subrotina chama a subrotina
* esforcos_concreto para calcular a parcela dos esforços resistentes devido ao
* concreto, e chama a subrotina tensao_aco para poder calcular a parcela dos
* esforços resistentes devido às barras de aço; por fim, soma as duas parcelas.
*Unidades de entrada: Sistema de unidades qualquer, desde que consistente;
*N_POLIG_CONCR e N_BARRAS_ACO: Número de nós da poligonal de concreto (incluindo
* o nó de chegada, que é o mesmo nó de saída) e número de barras de aço da
* seção;
*U_POLIG_CONCR, V_POLIG_CONCR e U_BARRAS_ACO,V_BARRAS_ACO: Vetores das
* coordenadas (sistema Ouv, ou seja, rotacionado em relação ao sistema Oxy) dos
* nós da seção de concreto (a poligonal deve ser numerada no sentido
* anti-horário, com vazios no sentido horário; com o nó de chegada) e das
* coordenadas das barras de aço (sistema Ouv);
*A_BARRAS_ACO: Vetor das áreas das barras de aço;
*N_m: Número de curvas (polinômios de grau qualquer) que compoem o diagrama
* tensão-deformação adotado para o concreto; cada curva é definida por um
* somatório do tipo: sigma_c=fc(C0.e^0 + C1.e^1 + C2.e^2 +...+Cm.e^m); para o
* diagrama parábola-retângulo da NB1/78, temos N_m=2 (curva parabólica e curva
* reta constante);
*VETOR_m: Vetor que contém os graus dos N_m polinômios;para o parábola-retângulo
* da NB1/78, temos VETOR_m=(2,0) (m=2 -> parábola; m=0 -> reta constante);
*MATRIZ_C: Matriz que contém os coeficientes dos N_m polinômios: linha i contém
* os Ck (k=1 à m+1) coeficientes do polinômio i; para o parábola-retângulo da
* NB1/78, temos: MATRIZ_C = !0 1 -0.25! -> coef. da parábola
* !1 0 0 ! -> coef. da reta constante;
*MATRIZ_e_LIMITES: Matriz que contém as deformações (em por mil) iniciais e
* finais onde age cada um dos N_m polinômios (deformações estas sem o efeito da
* fluência): linha i contém as deformações iniciais e finais onde age o
* polinômio i; OBS: colocar um número grande para a deformação final do último
* polinômio que compõe o diagrama tensão-deformação adotado para o concreto,
* não finalizando o diagrama no ELU; para o parábola-retângulo da NB1/78,temos:
* MATRIZ_e_LIMITES = !0 2 ! -> def. limites da parábola
* !2 1000! -> def. limites da reta constante;
*fc: Tensão máxima de compressão à que o concreto pode estar sujeito, ou seja,
* é a tensão máxima do diagrama tensão-deformação adotado para o concreto.
* Não há censo quanto ao valor a se utilizar. A NB1/78 indica fc=0.85fcd
* (ELU), mas, para o estudo da estabilidade, alguns autores indicam fc=0.85fck.
*FI_FLUENCIA: Função de Fluência (Teoria Linear de Fluência);

```

```

*fy: Tensão de escoamento do aço (valor de cálculo ou não);
*Es: Módulo de Elasticidade do aço (usualmente igual a 210000MPa);
*CLASSE_ACO: É a classe à que pertence o aço (A ou B);
*e_CG: Deformação (em por mil) do ponto de coordenadas (0,0), onde se admite
* localizar o centro de gravidade da poligonal; é em relação a este ponto que
* são calculados os esforços resistentes da seção; e_cg>0 -> encurtamento;
*TETA: Curvatura majorada adimensional (1000h/r); teta pode ser positivo.
* negativo ou zero;
*H_ALFA: Altura máxima da poligonal total de concreto (paralelam. ao eixo V);
*Nr ,Mru ,Mrv : Saídas da subrotina: Esforços resistentes totais da seção:
* Nr >0 --> força normal de compressão;
* Mru>0 --> momento definido por uma excentricidade positiva sobre o eixo u;
* Mrv>0 --> momento definido por uma excentricidade positiva sobre o eixo v.
*-----
* SISTEMA FLEXOR - DIMENSIONAMENTO E VERIFICAÇÃO DE PILARES ESBELTOS DE
* CONCRETO ARMADO COM SEÇÃO QUALQUER SOLICITADOS POR FLEXÃO
* COMPOSTA OBLÍQUA
* DIAG. TENSÃOxDEFORM.DO CONCRETO: POLINÔMIOS DE GRAU QUALQUER
* CONSIDERAÇÃO DA FLUÊNCIA: TEORIA LINEAR DE FLUÊNCIA
* AÇOS BRASILEIROS DO TIPO 'A' OU 'B' COM QUALQUER fy OU Es
* Copyright(c) 1997, Ismael Wilson Cadamuro Júnior
*-----
SUBROUTINE ESFORCOS_SECAO (N_POLIG_CONCR,N_BARRAS_ACO,
+ U_POLIG_CONCR,V_POLIG_CONCR,U_BARRAS_ACO,V_BARRAS_ACO,
+ A_BARRAS_ACO,N_m,VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,fc,
+ FI_FLUENCIA,fy,Es,CLASSE_ACO,e_CG,TETA,H_ALFA,Nr,Mru,Mrv)
CHARACTER*1 CLASSE_ACO
INTEGER N_POLIG_CONCR,N_BARRAS_ACO,N_m,VETOR_m(N_m+1),I
REAL*8 U_POLIG_CONCR(N_POLIG_CONCR+1),V_POLIG_CONCR(N_POLIG_CONCR+
+ 1),U_BARRAS_ACO(N_BARRAS_ACO+1),V_BARRAS_ACO(N_BARRAS_ACO+1),
+ A_BARRAS_ACO(N_BARRAS_ACO+1),MATRIZ_C(N_m+1,30),
+ MATRIZ_e_LIMITES(N_m+1,2),fc,FI_FLUENCIA,fy,Es,e_CG,TETA,Nr,Mru,
+ Mrv,Nrc,Mruc,Mrvc,Nrs,Mrus,Mrvs,e_s,SIGMA_S,F_S,H_ALFA
* Obs.: a dimensão dos vetores contém o "+1" para não ocorrer erro no caso
* de se entrar com n_polig_concr ou n_barras_aco igual a zero!
Nrc =0
Mruc=0
Mrvc=0
!cálc. dos esforços resistentes do concreto
CALL ESFORCOS_CONCRETO (N_POLIG_CONCR,U_POLIG_CONCR,V_POLIG_CONCR,
+ e_CG,TETA,N_m,VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,fc,FI_FLUENCIA
+ ,H_ALFA,Nrc,Mruc,Mrvc)
Nrs =0
Mrus=0
Mrvs=0
I=1
DO WHILE (I.LE.N_BARRAS_ACO) !cálc. dos esforços resistentes do aço
e_s=e_CG+TETA/H_ALFA*V_BARRAS_ACO(I) !deformação em cada barra
CALL TENSAO_ACO (e_s,fy,Es,CLASSE_ACO,SIGMA_S) !tensão em cada barra
F_S=SIGMA_S*A_BARRAS_ACO(I) !força normal em cada barra
Nrs=Nrs+F_S !soma dos esforços de cada barras de aço
Mrus=Mrus+U_BARRAS_ACO(I)*F_S
Mrvs=Mrvs+V_BARRAS_ACO(I)*F_S
I=I+1
END DO
Nr =Nrc +Nrs !cálc. dos esforços resistentes totais
Mru=Mruc+Mrus
Mrv=Mrvc+Mrvs
END
*=====fim da subrotina ESFORCOS_SECAO=====

*=====SUBROTINA TENSAO_ACO=====
*Subrotina que calcula a tensão em uma barra de aço.
*ATENÇÃO: Esta subrotina está adaptada aos aços e às curvas tensão-deformação
* definidas pela norma NB1/78;
*Unidades de entrada: Sistema de unidades qualquer, desde que consistente;
*e_s: Deformação (em por mil) do aço;
*fy: Tensão de escoamento do aço (valor de cálculo ou não);
*Es: Módulo de Elasticidade do aço (usualmente igual a 210000MPa);
*CLASSE_ACO: É à classe a que pertence o aço (A ou B);
*SIGMA_S: Saída da subrotina:Tensão na barra de aço (terá o mesmo sinal de e_s).
*-----
* SISTEMA FLEXOR - DIMENSIONAMENTO E VERIFICAÇÃO DE PILARES ESBELTOS DE

```

```

*           CONCRETO ARMADO COM SEÇÃO QUALQUER SOLICITADOS POR FLEXÃO
*           COMPOSTA OBLÍQUA
*           DIAG. TENSÃOxDEFORM.DO CONCRETO: POLINÔMIOS DE GRAU QUALQUER
*           CONSIDERAÇÃO DA FLUÊNCIA: TEORIA LINEAR DE FLUÊNCIA
*           AÇOS BRASILEIROS DO TIPO 'A' OU 'B' COM QUALQUER fy OU Es
*           Copyright(c) 1997, Ismael Wilson Cadamuro Júnior
*-----

```

```

SUBROUTINE TENSAO_ACO (e_s,fy,Es,CLASSE_ACO,SIGMA_S)
CHARACTER*1 CLASSE_ACO
REAL*8 e_s,fy,Es,SIGMA_S,e2_s,eyd
IF (e_s.GE.0) THEN                                !guardando o sinal da deformação
  SINAL=+1
ELSE
  SINAL=-1
END IF
e2_s=ABS(e_s)                                     !deformação em módulo para entrada nas fórmulas
IF (CLASSE_ACO.EQ.'A') THEN                       !cálc. de eyd
  eyd=1000*fy/Es
ELSE
  IF (CLASSE_ACO.EQ.'B') THEN
    eyd=1000*fy/Es+2
  ELSE
    CALL BEEPQQ(1500,1000)
    CALL BEEPQQ(1500,1000)
    WRITE(*,*) 'ERRO (SUBROTINA TENSAO_ACO): CLASSE DO ACO DIFE',
+ 'RENTE DE A OU B'                               !teste de consistência
    STOP
  END IF
END IF
IF (e2_s.LE.0.7*1000*fy/Es) THEN                  !cálc. da tensão (em módulo)
  SIGMA_S=Es*e2_s/1000
ELSE
  IF (e2_s.GE.eyd) THEN
    SIGMA_S=fy
  ELSE
    IF (CLASSE_ACO.EQ.'A') THEN
      SIGMA_S=Es*e2_s/1000
    ELSE
      SIGMA_S=fy*(0.7-22.5*fy*(1/Es-SQRT(1/Es**2+
+ 4./45/fy*(e2_s/fy/1000-0.7/Es))))
    END IF
  END IF
END IF
SIGMA_S=SIGMA_S*SINAL                             !cálc. da tensão (com sinal)
END

```

```

*=====fim da subrotina TENSAO_ACO=====

```

```

*=====SUBROTINA ESFORCOS_CONCRETO=====

```

```

*Subrotina que calcula os esforços resistentes do concreto para uma seção com
* geometria e estado de deformação dados. Admite-se que a linha neutra seja
* paralela ao eixo das abscissas (u). Esta subrotina chama a subrotina
* extrai_poligonal para calcular a(s) poligonal(is) comprimida(s) do concreto,
* e chama a subrotina integral_concreto para calcular os esforços resistentes
* desta(s) poligonal(is) comprimida(s). Caso o !teta! dado esteja entre 0 e
* 1E-3, esta subrotina chama as subrotinas integral_concreto_teta_zero e
* integral_concreto e faz uma interpolação linear para calcular os resultados
* finais.
*Unidades de entrada: Sistema de unidades qualquer, desde que consistente;
*N_POLIG: Número de nós da poligonal de concreto (incluindo o nó de chegada, que
* deve ter as mesmas coordenadas do nó de saída);
*U_POLIG e V_POLIG: Vetores das coordenadas dos nós da seção de concreto
* (sistema Ouv); a poligonal deve ser numerada no sentido anti-horário
* (vazios -> sentido horário);
*e_CG: Deformação (em por mil) do ponto de coordenadas (0,0), onde se admite
* localizar o centro de gravidade da poligonal; é em relação a este ponto
* que são calculados os esforços resistentes da seção.
* e_cg>0 -> encurtamento;
*TETA: Curvatura majorada adimensional (1000h/r); teta pode ser positivo,
* negativo ou zero;
*N_m: Número de curvas (polinômios de grau qualquer) que compoem o diagrama
* tensão-deformação adotado para o concreto; cada curva é definida por um
* somatório do tipo: sigma_c=fc(C0.e^0 + C1.e^1 + C2.e^2 +...+Cm.e^m); para o

```

```

* diagrama parábola-retângulo da NBl/78, temos N_m=2 (curva parabólica e curva
* reta constante);
*VETOR_m: Vetor que contém os graus dos N_m polinômios;para o parábola-retângulo
* da NBl/78, temos VETOR_m=(2,0) (m=2 -> parábola; m=0 -> reta constante);
*MATRIZ_C: Matriz que contém os coeficientes dos N_m polinômios: linha i contém
* os Ck (k=1 à m+1) coeficientes do polinômio i; para o parábola-retângulo da
* NBl/78, temos: MATRIZ_C = !0 1 -0.25! -> coef. da parábola
* !1 0 0 ! -> coef. da reta constante;
*MATRIZ_e_LIMITES: Matriz que contém as deformações (em por mil) iniciais e
* finais onde age cada um dos N_m polinômios (deformações estas sem o efeito da
* fluência): linha i contém as deformações iniciais e finais onde age o
* polinômio i; OBS: colocar um número grande para a deformação final do último
* polinômio que compõe o diagrama tensão-deformação adotado para o concreto,
* não finalizando o diagrama no ELU; para o parábola-retângulo da NBl/78,temos:
* MATRIZ_e_LIMITES = !0 2 ! -> def. limites da parábola
* !2 1000! -> def. limites da reta constante;
*fc: Tensão máxima de compressão à que o concreto pode estar sujeito, ou seja,
* é a tensão máxima do diagrama tensão-deformação adotado para o concreto.
* Não há censo quanto ao valor a se utilizar. A NBl/78 indica fc=0.85fcd
* (ELU), mas, para o estudo da estabilidade, alguns autores indicam fc=0.85fck.
*FI_FLUENCIA: Função de Fluência (Teoria Linear de Fluência);
*H_ALFA: Altura máxima da poligonal total de concreto (paralelam. ao eixo V);
*Nrc,Mruc,Mrvc: Saídas da subrotina: Esforços resistentes do concreto da seção;
* Nrc >0 --> força normal de compressão;
* Mruc>0 --> momento definido por uma excentricidade positiva sobre o eixo u;
* Mrvc>0 --> momento definido por uma excentricidade positiva sobre o eixo v.
*-----
* SISTEMA FLEXOR - DIMENSIONAMENTO E VERIFICAÇÃO DE PILARES ESBELTOS DE
* CONCRETO ARMADO COM SEÇÃO QUALQUER SOLICITADOS POR FLEXÃO
* COMPOSTA OBLÍQUA
* DIAG. TENSÃOxDEFORM.DO CONCRETO: POLINÔMIOS DE GRAU QUALQUER
* CONSIDERAÇÃO DA FLUÊNCIA: TEORIA LINEAR DE FLUÊNCIA
* AÇOS BRASILEIROS DO TIPO 'A' OU 'B' COM QUALQUER fy OU Es
* Copyright(c) 1997, Ismael Wilson Cadamuro Júnior
*-----

```

```

SUBROUTINE ESFORCOS_CONCRETO (N_POLIG,U_POLIG,V_POLIG,e_CG,TETA,
+ N_m,VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,fc,FI_FLUENCIA,H_ALFA,
+ Nrc,Mruc,Mrvc)
INTEGER N_POLIG,N_POLIG_COMPR,N_m,VETOR_m(N_m+1),I,K
REAL*8 U_POLIG(N_POLIG+1),V_POLIG(N_POLIG+1),e_CG,TETA,
+ MATRIZ_C(N_m+1,30),MATRIZ_e_LIMITES(N_m+1,2),fc,FI_FLUENCIA,
+ Nrc,Mruc,Mrvc,H_ALFA,U_POLIG_COMPR[ALLOCATABLE](:),
+ V_POLIG_COMPR[ALLOCATABLE](:),C(30),Nrc1,Mruc1,Mrvc1,V_CORT1,
+ V_CORT2,Nrc_TETA_0,Mruc_TETA_0,Mrvc_TETA_0,Nrc_TETA_3,
+ Mruc_TETA_3,Mrvc_TETA_3,TETA_3,SINAL
ALLOCATE(U_POLIG_COMPR(2*N_POLIG+1),V_POLIG_COMPR(2*N_POLIG+1))
* Obs.: a dimensão dos vetores contém o "+1" para não ocorrer erro no caso
* de se entrar com n_polig=0; os "2*n_polig" foram usados porque se estima
* que essa poligonal jamais terá mais que o dobro de nós da poligonal a
* ser cortada (este artifício foi utilizado porque não se sabe de início
* qual o número de nós desta poligonal).
IF (N_POLIG.EQ.0) GOTO 1 !não há poligonal
IF (N_POLIG.LE.3) THEN !teste de consistência
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) 'ERRO (SUBROTINA ESFORCOS_CONCRETO): POLIGONAL COM',
+ 'MENOS DE 4 NOS'
STOP
END IF
IF (H_ALFA.EQ.0) THEN !teste de consistência
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) 'ERRO (SUBROTINA ESFORCOS_CONCRETO): ALTURA DA ',
+ 'SECAO IGUAL A ZERO'
STOP
END IF !caso teta<1E-3: chamar integral_concreto_teta_zero
IF (ABS(TETA).LT.1E-3) THEN!e integral_concreto e interpolar os resultados
CALL INTEGRAL_CONCRETO_TETA_ZERO (e_CG,N_POLIG,U_POLIG,V_POLIG,
+ N_m,VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,fc,FI_FLUENCIA,
+ Nrc_TETA_0,Mruc_TETA_0,Mrvc_TETA_0)
IF (TETA.GE.0) THEN
SINAL=+1
ELSE
SINAL=-1

```

```

END IF
TETA_3=SINAL*1E-3
Nrc_TETA_3=0
Mruc_TETA_3=0
Mrvc_TETA_3=0
I=1
DO WHILE (I.LE.N_m)!loop para cada curva do diagr.tensão-def.do concreto
m=VETOR_m(I)      !grau do polinômio que define a curva i do diagrama
K=1
DO WHILE (K.LE.VETOR_m(I)+1)      !cálc. do vetor dos coef. da curva i
C(K)=MATRIZ_C(I,K)
K=K+1
END DO      !cálc.das ordenadas limites onde age a curva i
V_CORT1=(MATRIZ_e_LIMITES(I,1)*(1+FI_FLUENCIA)-e_CG)*H_ALFA/
+ TETA_3
V_CORT2=(MATRIZ_e_LIMITES(I,2)*(1+FI_FLUENCIA)-e_CG)*H_ALFA/
+ TETA_3
CALL EXTRAI_POLIGONAL (N_POLIG,U_POLIG,V_POLIG,V_CORT1,V_CORT2
+ ,N_POLIG_COMPR,U_POLIG_COMPR,V_POLIG_COMPR)      !cálc.polig.compr.
CALL INTEGRAL_CONCRETO(N_POLIG_COMPR,U_POLIG_COMPR,
+ V_POLIG_COMPR,e_CG,TETA_3,H_ALFA,FI_FLUENCIA,m,C,fc,
+ Nrcl,Mruc1,Mrvc1)
Nrc_TETA_3=Nrc_TETA_3+Nrc1      !cálc.esforços
Mruc_TETA_3=Mruc_TETA_3+Mruc1      !soma das contribuições
Mrvc_TETA_3=Mrvc_TETA_3+Mrvc1
I=I+1
END DO      !interpolação linear dos resultados
Nrc=Nrc_TETA_0+(Nrc_TETA_3-Nrc_TETA_0)*TETA/TETA_3
Mruc=Mruc_TETA_0+(Mruc_TETA_3-Mruc_TETA_0)*TETA/TETA_3
Mrvc=Mrvc_TETA_0+(Mrvc_TETA_3-Mrvc_TETA_0)*TETA/TETA_3
GOTO 1
END IF
Nrc=0
Mruc=0
Mrvc=0
I=1      !caso teta>1E-3, continua assim...
DO WHILE (I.LE.N_m) !loop para cada curva do diagr.tensão-def.do concreto
m=VETOR_m(I)      !grau do polinômio que define a curva i do diagrama
K=1
DO WHILE (K.LE.VETOR_m(I)+1)      !cálc. do vetor dos coef. da curva i
C(K)=MATRIZ_C(I,K)
K=K+1
END DO      !cálc.das ordenadas limites onde age a curva i
V_CORT1=(MATRIZ_e_LIMITES(I,1)*(1+FI_FLUENCIA)-e_CG)*H_ALFA/TETA
V_CORT2=(MATRIZ_e_LIMITES(I,2)*(1+FI_FLUENCIA)-e_CG)*H_ALFA/TETA
CALL EXTRAI_POLIGONAL (N_POLIG,U_POLIG,V_POLIG,V_CORT1,V_CORT2,
+ N_POLIG_COMPR,U_POLIG_COMPR,V_POLIG_COMPR)      !cálc.polig.compr.
CALL INTEGRAL_CONCRETO(N_POLIG_COMPR,U_POLIG_COMPR,V_POLIG_COMPR
+ ,e_CG,TETA,H_ALFA,FI_FLUENCIA,m,C,fc,Nrcl,Mruc1,Mrvc1)
Nrc=Nrc+Nrc1      !cálc.esforços
Mruc=Mruc+Mruc1      !soma das contribuições
Mrvc=Mrvc+Mrvc1
I=I+1
END DO
1 DEALLOCATE(U_POLIG_COMPR,V_POLIG_COMPR)
END
=====fim da subrotina ESFORCOS_CONCRETO=====

```

```

=====SUBROTINA MAXIMO_MINIMO_DIFEREN=====
*Subrotina que calcula os valores máximo e mínimo de um vetor dado, e a
* diferença entre estes valores.
*N_VETOR: Número de dados do vetor.
*VETOR: Vetor do qual se quer os valores máximos e mínimos.
*MAXIM e MINIM: Saídas da subrotina: São os valores máximos e mínimos do vetor.
*DIFERENCA: Saída da subrotina: É a diferença entre maxim e minim.
*-----
* SISTEMA FLEXOR - DIMENSIONAMENTO E VERIFICAÇÃO DE PILARES ESBELTOS DE
* CONCRETO ARMADO COM SEÇÃO QUALQUER SOLICITADOS POR FLEXÃO
* COMPOSTA OBLÍQUA
* DIAG. TENSÃOxDEFORM.DO CONCRETO: POLINÔMIOS DE GRAU QUALQUER
* CONSIDERAÇÃO DA FLUÊNCIA: TEORIA LINEAR DE FLUÊNCIA
* AÇOS BRASILEIROS DO TIPO 'A' OU 'B' COM QUALQUER fy OU Es

```



```

* Copyright(c) 1997, Ismael Wilson Cadamuro Júnior
*-----
      SUBROUTINE MAXIMO_MINIMO_DIFEREN (N_VETOR,VETOR,MAXIM,MINIM,
+ DIFERENCA)
      INTEGER N_VETOR,I
      REAL*8 VETOR(N_VETOR+1),MAXIM,MINIM,DIFERENCA
      MAXIM=0          !obs: dimensão do vetor contém o "+1" para não ocorrer
      MINIM=0          !          erro no caso de se entrar com n_vetor=0
      DIFERENCA=0
      IF (N_VETOR.EQ.0) GOTO 1
      MAXIM=VETOR(1)
      MINIM=VETOR(1)
      I=2
      DO WHILE (I.LE.N_VETOR)
          IF (VETOR(I).GT.MAXIM) MAXIM=VETOR(I)
          IF (VETOR(I).LT.MINIM) MINIM=VETOR(I)
          I=I+1
      END DO
      DIFERENCA=MAXIM-MINIM
1  END
*=====fim da subrotina MAXIMO_MINIMO_DIFEREN=====

*=====SUBROTINA EXTRAI_POLIGONAL=====
*Subrotina que, dada uma poligonal origem, extrai desta a parte que estiver
* entre as ordenadas v_corte1 e v_corte2, montando uma poligonal resultado.
* Esta subrotina é utilizada na montagem da(s) poligonal(ais) comprimida(s) de
* concreto.
* Esta subrotina pode resultar em poligonais degeneradas à 0 nós (quando não há
* poligonal entre v_corte1 e v_corte2).Também pode resultar em poligonais com
* lados de nós inicial e final de mesma coordenada.
*Unidades de entrada: Sistema de unidades qualquer, desde que consistente;
*N_POLIG: Número de nós da poligonal original (incluindo o nó de chegada,que
* deve ter as mesmas coordenadas do nó de saída);
*U_POLIG e V_POLIG: Vetores das coordenadas dos nós da poligonal original
* (sistema Ouv);
*V_CORTE1 e V_CORTE2: Ordenadas entre as quais será extraída a poligonal
* resultado; não há restrição em relação a qual é o maior ou o menor valor;
*N_POLIG_RESUL:Saída da subrotina: número de nós da poligonal resultado
* (incluindo o nó de chegada, que é o mesmo nó de saída);
*U_POLIG_RESUL,V_POLIG_RESUL:Saídas da subrotina:Vetores das coordenadas dos nós
* da poligonal resultado.
*-----
* SISTEMA FLEXOR - DIMENSIONAMENTO E VERIFICAÇÃO DE PILARES ESBELTOS DE
* CONCRETO ARMADO COM SEÇÃO QUALQUER SOLICITADOS POR FLEXÃO
* COMPOSTA OBLÍQUA
* DIAG. TENSÃOxDEFORM.DO CONCRETO: POLINÔMIOS DE GRAU QUALQUER
* CONSIDERAÇÃO DA FLUÊNCIA: TEORIA LINEAR DE FLUÊNCIA
* AÇOS BRASILEIROS DO TIPO 'A' OU 'B' COM QUALQUER fy OU Es
* Copyright(c) 1997, Ismael Wilson Cadamuro Júnior
*-----
      SUBROUTINE EXTRAI_POLIGONAL (N_POLIG,U_POLIG,V_POLIG,V_CORTE1,
+ V_CORTE2,N_POLIG_RESUL,U_POLIG_RESUL,V_POLIG_RESUL)
      INTEGER N_POLIG,N_POLIG_RESUL,POSICAO_NO[ALLOCATABLE](:),I,K
      REAL*8 U_POLIG(N_POLIG+1),V_POLIG(N_POLIG+1),V_CORTE1,V_CORTE2,
+ U_POLIG_RESUL(2*N_POLIG+1),V_POLIG_RESUL(2*N_POLIG+1),V_MAX,
+ V_MIN,V_CORTE_MAX,V_CORTE_MIN,AUX
* Obs.: a dimensão dos vetores contém o "+1" para não ocorrer erro no caso
* de se entrar com n_polig=0; os "2*n_polig" foram usados porque se estima
* que essa poligonal jamais terá mais que o dobro de nós da poligonal a
* ser cortada (este artifício foi utilizado porque não se sabe de início
* qual o número de nós da poligonal resultante).
      ALLOCATE (POSICAO_NO(N_POLIG+1))
      N_POLIG_RESUL=0
      IF (N_POLIG.EQ.0) GOTO 1          !não há poligonal original
      IF (V_CORTE1.EQ.V_CORTE2) GOTO 1 !não há poligonal resultado
      IF (N_POLIG.LE.3) THEN          !teste de consistência
          CALL BEEPQQ(1500,1000)
          CALL BEEPQQ(1500,1000)
          WRITE(*,*) 'ERRO (SUBROTINA EXTRAI_POLIGONAL):POLIGONAL COM ',
+ 'MENOS DE 4 NOS'
          STOP
      END IF

```

```

IF (V_CORTE1.LE.V_CORTE2) THEN !cálc. da maior e menor ordenadas de corte
  V_CORTE_MIN=V_CORTE1
  V_CORTE_MAX=V_CORTE2
ELSE
  V_CORTE_MIN=V_CORTE2
  V_CORTE_MAX=V_CORTE1
END IF
CALL MAXIMO_MINIMO_DIFEREN (N_POLIG,V_POLIG,V_MAX,V_MIN,AUX)
IF (V_CORTE_MAX.LE.V_MIN) GOTO 1 !não há poligonal no meio das
IF (V_CORTE_MIN.GE.V_MAX) GOTO 1 ! duas ordenadas de corte
I=1
DO WHILE (I.LE.N_POLIG)
  IF (V_POLIG(I).GT.V_CORTE_MAX) THEN !variável de auxílio
    POSICAO_NO(I)=1 ! na localização de
  ELSE ! cada nó
    IF (V_POLIG(I).LT.V_CORTE_MIN) THEN
      POSICAO_NO(I)=-1
    ELSE
      POSICAO_NO(I)=0
    END IF
  END IF
  I=I+1
END DO
K=0
I=1
DO WHILE (I.LE.N_POLIG-1) !loop para cada lado da poligonal
  IF (POSICAO_NO(I).EQ.0) THEN !colocando o nó i do lado i na poligonal
    K=K+1
    U_POLIG_RESUL(K)=U_POLIG(I)
    V_POLIG_RESUL(K)=V_POLIG(I)
  END IF
  IF (POSICAO_NO(I)+POSICAO_NO(I+1).EQ.-1) THEN !lado i atravessa
    U_CORTE_LADO=U_POLIG(I)-(V_POLIG(I)-V_CORTE_MIN)* ! v_corte_min
    + (U_POLIG(I+1)-U_POLIG(I))/(V_POLIG(I+1)-V_POLIG(I))
    K=K+1
    U_POLIG_RESUL(K)=U_CORTE_LADO !colocando o nó do corte
    V_POLIG_RESUL(K)=V_CORTE_MIN ! na poligonal resultado
  END IF
  IF (POSICAO_NO(I)+POSICAO_NO(I+1).EQ.+1) THEN !lado i atravessa
    U_CORTE_LADO=U_POLIG(I)-(V_POLIG(I)-V_CORTE_MAX)* ! v_corte_max
    + (U_POLIG(I+1)-U_POLIG(I))/(V_POLIG(I+1)-V_POLIG(I))
    K=K+1
    U_POLIG_RESUL(K)=U_CORTE_LADO !colocando o nó do corte
    V_POLIG_RESUL(K)=V_CORTE_MAX ! na poligonal resultado
  END IF
  IF (POSICAO_NO(I)*POSICAO_NO(I+1).EQ.-1) THEN !lado i atravessa ambas
    IF (POSICAO_NO(I).EQ.-1) THEN ! as ordenadas de corte
      U_CORTE_LADO=U_POLIG(I)-(V_POLIG(I)-V_CORTE_MIN)*
      + (U_POLIG(I+1)-U_POLIG(I))/(V_POLIG(I+1)-V_POLIG(I))
      K=K+1
      U_POLIG_RESUL(K)=U_CORTE_LADO !de baixo para cima, ou...
      V_POLIG_RESUL(K)=V_CORTE_MIN
      U_CORTE_LADO=U_POLIG(I)-(V_POLIG(I)-V_CORTE_MAX)*
      + (U_POLIG(I+1)-U_POLIG(I))/(V_POLIG(I+1)-V_POLIG(I))
      K=K+1
      U_POLIG_RESUL(K)=U_CORTE_LADO
      V_POLIG_RESUL(K)=V_CORTE_MAX
    ELSE
      U_CORTE_LADO=U_POLIG(I)-(V_POLIG(I)-V_CORTE_MAX)*
      + (U_POLIG(I+1)-U_POLIG(I))/(V_POLIG(I+1)-V_POLIG(I))
      K=K+1
      U_POLIG_RESUL(K)=U_CORTE_LADO !de cima para baixo.
      V_POLIG_RESUL(K)=V_CORTE_MAX
      U_CORTE_LADO=U_POLIG(I)-(V_POLIG(I)-V_CORTE_MIN)*
      + (U_POLIG(I+1)-U_POLIG(I))/(V_POLIG(I+1)-V_POLIG(I))
      K=K+1
      U_POLIG_RESUL(K)=U_CORTE_LADO
      V_POLIG_RESUL(K)=V_CORTE_MIN
    END IF
  END IF
  I=I+1
END DO
K=K+1 !fechando a poligonal resultado
U_POLIG_RESUL(K)=U_POLIG_RESUL(1)

```

```

V_POLIG_RESUL(K)=V_POLIG_RESUL(1)
N_POLIG_RESUL=K          !número total de nós da poligonal resultado
1  DEALLOCATE (POSICAO_NO)
END
*=====fim da subrotina EXTRAI_POLIGONAL=====

*=====SUBROTINA INTEGRAL_CONCRETO_TETA_ZERO=====
*Subrotina que calcula os esforços resistentes do concreto (Nrc,Mruc e Mrvc)
* quando se tem teta (curvatura majorada adimensional: 1000h/r) igual a zero.
* Esta subrotina se faz necessária porque no caso de TETA=0, a subrotina
* integral_concreto não pode ser utilizada (teta está no divisor de algumas
* fórmulas). Admite-se que o diagrama tensão-deformação do concreto comprimido
* seja formado por N_m curvas polinomiais de grau qualquer.
*Unidades de entrada: Sistema de unidades qualquer, desde que consistente;
*e_SECAO: Deformação (em por mil) à que a seção está submetida (como teta=0,
* esta deformação age sobre todos os pontos da seção); deformação de
* encurtamento -> e_secao positiva;
*N_POLIG: Número de nós da poligonal comprimida (incluindo o nó de chegada, que
* deve ter as mesmas coordenadas do nó de saída);
*U_POLIG e V_POLIG: Vetores das coordenadas dos nós da poligonal comprimida
* (sistema Ouv); a poligonal deve ser numerada no sentido anti-horário
* (vazios -> sentido horário);
*N_m: Número de curvas (polinômios de grau qualquer) que compoem o diagrama
* tensão-deformação adotado para o concreto; cada curva é definida por um
* somatório do tipo: sigma_c=fc(C0.e^0 + C1.e^1 + C2.e^2 +...+Cm.e^m); para o
* diagrama parábola-retângulo da NB1/78, temos N_m=2 (curva parabólica e curva
* reta constante);
*VETOR_m: Vetor que contém os graus dos N_m polinômios;para o parábola-retângulo
* da NB1/78, temos VETOR_m=(2,0) (m=2 -> parábola; m=0 -> reta constante);
*MATRIZ_C: Matriz que contém os coeficientes dos N_m polinômios: linha i contém
* os Ck (k=1 à m+1) coeficientes do polinômio i; para o parábola-retângulo da
* NB1/78, temos: MATRIZ_C = !0 1 -0.25! -> coef. da parábola
* !1 0 0 ! -> coef. da reta constante;
*MATRIZ_e_LIMITES: Matriz que contém as deformações (em por mil) iniciais e
* finais onde age cada um dos N_m polinômios (deformações estas sem o efeito da
* fluência): linha i contém as deformações iniciais e finais onde age o
* polinômio i; OBS: colocar um número grande para a deformação final do último
* polinômio que compõe o diagrama tensão-deformação adotado para o concreto,
* não finalizando o diagrama no ELU; para o parábola-retângulo da NB1/78,temos:
* MATRIZ_e_LIMITES = !0 2 ! -> def. limites da parábola
* !2 1000! -> def. limites da reta constante;
*fc: Tensão máxima de compressão à que o concreto pode estar sujeito, ou seja,
* é a tensão máxima do diagrama tensão-deformação adotado para o concreto.
* Não há consenso quanto ao valor a se utilizar. A NB1/78 indica fc=0.85fcd
* (ELU), mas, para o estudo da estabilidade, alguns autores indicam fc=0.85fck.
*FI_FLUENCIA: Função de Fluência (Teoria Linear de Fluência);
*Nrc,Mruc,Mrvc: Esforços resistentes do concreto (dados de saída da subrotina);
* Nrc >0 --> força normal de compressão;
* Mruc>0 --> momento definido por uma excentricidade positiva sobre o eixo U;
* Mrvc>0 --> momento definido por uma excentricidade positiva sobre o eixo V.
*-----
* SISTEMA FLEXOR - DIMENSIONAMENTO E VERIFICAÇÃO DE PILARES ESBELTOS DE
* CONCRETO ARMADO COM SEÇÃO QUALQUER SOLICITADOS POR FLEXÃO
* COMPOSTA OBLÍQUA
* DIAG. TENSÃOxDEFORM.DO CONCRETO: POLINÔMIOS DE GRAU QUALQUER
* CONSIDERAÇÃO DA FLUÊNCIA: TEORIA LINEAR DE FLUÊNCIA
* AÇOS BRASILEIROS DO TIPO 'A' OU 'B' COM QUALQUER fy OU Es
* Copyright(c) 1997, Ismael Wilson Cadamuro Júnior
*-----
SUBROUTINE INTEGRAL_CONCRETO_TETA_ZERO (e_SECAO,N_POLIG,U_POLIG,
+ V_POLIG,N_m,VETOR_m,MATRIZ_C,MATRIZ_e_LIMITES,fc,FI_FLUENCIA,
+ Nrc,Mruc,Mrvc)
INTEGER N_POLIG,N_m,VETOR_m(N_m+1),CURVA,I,K
REAL*8 U_POLIG(N_POLIG+1),V_POLIG(N_POLIG+1),e_SECAO,
+ MATRIZ_C(N_m+1,30),MATRIZ_e_LIMITES(N_m+1,2),fc,FI_FLUENCIA,
+ Nrc,Mruc,Mrvc,e_1,e_2,SIGMA_C,AREA,U_CG,V_CG
Nrc =0 !obs: dimensão dos vetores U e V contém
Mruc=0 ! o "+1" para não ocorrer erro no
Mrvc=0 ! caso de se entrar com n_polig=0
IF (N_POLIG.EQ.0) GOTO 1 !não há poligonal comprimida
IF (N_POLIG.LE.3) THEN !teste de consistência
CALL BEEPQQ(1500,1000)

```

```

        CALL BEEPQQ(1500,1000)
        WRITE(*,*) 'ERRO (SUBROTINA INTEGRAL_CONCRETO_TETA_ZERO): ',
+ ' POLIGONAL COMPRIMIDA COM MENOS DE 4 NOS'
        STOP
    END IF
    CURVA=0
    I=1
    DO WHILE (I.LE.N_m)                !procura da curva à qual pertence e_secao
        e_1=MATRIZ_e_LIMITES(I,1)*(1+FI_FLUENCIA)                !domínio da curva i
        e_2=MATRIZ_e_LIMITES(I,2)*(1+FI_FLUENCIA)
        IF (e_SECAO.GE.e_1.AND.e_SECAO.LE.e_2) CURVA=I
        I=I+1
    END DO
    IF (CURVA.EQ.0) GOTO 1                !e_secao não induz tensão no concreto
    SIGMA_C=0
    K=0
    DO WHILE (K.LE.VETOR_m(CURVA))        !cálc. da tensão no concreto
        SIGMA_C=SIGMA_C+MATRIZ_C(CURVA,K+1)*e_SECAO**K/(1+FI_FLUENCIA)**K
        K=K+1
    END DO
    SIGMA_C=fc*SIGMA_C                !cálc. área e cg da poligonal comprimida
    CALL GEOMETRIA (N_POLIG,U_POLIG,V_POLIG,AREA,U_CG,V_CG)
    Nrc =AREA*SIGMA_C                !cálc. dos esforços resistentes
    Mruc=Nrc*U_CG                    ! do concreto
    Mrvc=Nrc*V_CG
1    END
=====fim da subrotina INTEGRAL_CONCRETO_TETA_ZERO=====

```

```

=====SUBROTINA GEOMETRIA=====
*Subrotina que, dada uma poligonal, calcula a área e as coordenadas do seu
* centro de gravidade.
*Unidades de entrada: Sistema de unidades qualquer, desde que consistente;
*N_POLIG: Número de nós da poligonal (incluindo o nó de chegada, que deve ter as
* mesmas coordenadas do nó de saída);
*X_POLIG e Y_POLIG: Vetores das coordenadas dos nós da poligonal; a poligonal
* deve ser numerada no sentido trigonométrico (anti-horário), com eventuais
* vazios no sentido horário;
*AREA: Área da poligonal;
*X_CG e Y_CG: Coordenadas do centro de gravidade da poligonal.
*-----
* SISTEMA FLEXOR - DIMENSIONAMENTO E VERIFICAÇÃO DE PILARES ESBELTOS DE
* CONCRETO ARMADO COM SEÇÃO QUALQUER SOLICITADOS POR FLEXÃO
* COMPOSTA OBLÍQUA
* DIAG. TENSÃOxDEFORM.DO CONCRETO: POLINÔMIOS DE GRAU QUALQUER
* CONSIDERAÇÃO DA FLUÊNCIA: TEORIA LINEAR DE FLUÊNCIA
* AÇOS BRASILEIROS DO TIPO 'A' OU 'B' COM QUALQUER fy OU Es
* Copyright(c) 1997, Ismael Wilson Cadamuro Júnior
*-----

```

```

SUBROUTINE GEOMETRIA (N_POLIG,X_POLIG,Y_POLIG,AREA,X_CG,Y_CG)
INTEGER N_POLIG,I
REAL*8 X_POLIG(N_POLIG+1),Y_POLIG(N_POLIG+1),AREA,X_CG,Y_CG,Sxx,
+ Syy,AUX                !obs: a dimensão dos vetores contém
AREA=0                    ! o "+1" para não ocorrer erro no
Sxx=0                    ! caso de se entrar com n_polig=0
Syy=0
X_CG=0
Y_CG=0
IF (N_POLIG.EQ.0) GOTO 1                !não há poligonal
IF (N_POLIG.LE.3) THEN                !teste de consistência
    CALL BEEPQQ(1500,1000)
    CALL BEEPQQ(1500,1000)
    WRITE(*,*) 'ERRO (SUBROTINA GEOMETRIA): POLIGONAL COM MENOS DE',
+ ' 4 NOS'
    STOP
END IF
IF (X_POLIG(1).NE.X_POLIG(N_POLIG).OR.                !teste de consistência
+ Y_POLIG(1).NE.Y_POLIG(N_POLIG)) THEN
    CALL BEEPQQ(1500,1000)
    CALL BEEPQQ(1500,1000)
    WRITE(*,*) 'ERRO (SUBROTINA GEOMETRIA): POLIGONAL COM NOS INI',
+ ' CIAL E FINAL COM COORDENADAS DIFERENTES'
    STOP

```

```

END IF
I=1
DO WHILE (I.LE.N_POLIG-1)
  AUX=X_POLIG(I)*Y_POLIG(I+1)-X_POLIG(I+1)*Y_POLIG(I)
  AREA=AREA+AUX
  Sxx=Sxx+AUX*(Y_POLIG(I)+Y_POLIG(I+1))
  Syy=Syy+AUX*(X_POLIG(I)+X_POLIG(I+1))
  I=I+1
END DO
AREA=AREA/2
Sxx=Sxx/6
Syy=Syy/6
X_CG=Syy/AREA
Y_CG=Sxx/AREA
IF (AREA.LT.0) THEN
  CALL BEEPQQ(1500,1000)
  CALL BEEPQQ(1500,1000)
  WRITE(*,*) 'ERRO (SUBROTINA GEOMETRIA): POLIGONAL COM AREA NE',
+ 'GATIVA --> A POLIGONAL NAO FOI NUMERADA NO SENTIDO TRIGONO',
+ 'METRICO (ANTI-HORARIO)'
  STOP
END IF
1 END
=====fim da subrotina GEOMETRIA=====

=====SUBROTINA INTEGRAL_CONCRETO=====
*Subrotina que calcula os esforços resistentes do concreto (Nrc,Mruc e Mrvc)
* através de uma integração analítica das tensões na poligonal comprimida dada.
* Admite-se diagrama tensão-deformação polinomial para o concreto comprimido.
* Admite-se que a linha neutra seja paralela ao eixo u. Esta subrotina pode ser
* utilizada com somente uma equação definindo o diagrama tensão-deformação do
* concreto; no caso do diagrama parábola-retângulo, deve-se chamar esta
* subrotina duas vezes: uma para a parte parabólica e outra para a parte
* constante, com as suas respectivas poligonais comprimidas.
*Unidades de entrada: Sistema de unidades qualquer, desde que consistente;
*N_POLIG: Número de nós da poligonal comprimida (incluindo o nó de chegada, que
* deve ter as mesmas coordenadas do nó de saída);
*U_POLIG e V_POLIG: Vetores das coordenadas dos nós da poligonal comprimida
* (sistema Ouv); a poligonal deve ser numerada no sentido anti-horário
* (vazios -> sentido horário);
*e_cg: Deformação (em por mil) do ponto de coordenadas (0,0), onde se admite
* localizar o centro de gravidade da poligonal total (não só a comprimida);
* e_cg>0 -> encurtamento;
*TETA: Curvatura majorada adimensional (1000h/r); para !teta! menor que
* aproximadamente 1E-6, começa a haver imprecisões numéricas nos resultados
* desta subrotina; se se entrar com !teta!<1E-6, a subrotina avisa este fato ao
* usuário e causa uma pausa; admite-se teta positivo ou negativo, mas não zero;
*H_ALFA: Altura máxima da poligonal total de concreto (paralelam. ao eixo V);
*FI_FLUENCIA: Função de Fluência (Teoria Linear de Fluência);
*m,C,fc: Parâmetros do diagrama tensão-deformação polinomial do concreto (sem o
* efeito da fluência). A curva é definida por um somatório do tipo:
* sigma_c=fc(C0.e^0 + C1.e^1 + C2.e^2 +...+Cm.e^m). Para o diagrama
* parábola-retângulo da NBl/78, temos:
* parte parabólica---> m=2; C=(0,1,-0.25); fc=0,85fcd (p/ ELU de ruptura);
* parte retangular---> m=0; C=(1); fc=0,85fcd (p/ ELU de ruptura);
*Nrc,Mruc,Mrvc: Esforços resistentes do concreto (dados de saída da subrotina);
* Nrc >0 --> força normal de compressão;
* Mruc>0 --> momento definido por uma excentricidade positiva sobre o eixo U;
* Mrvc>0 --> momento definido por uma excentricidade positiva sobre o eixo V.
-----
* SISTEMA FLEXOR - DIMENSIONAMENTO E VERIFICAÇÃO DE PILARES ESBELTOS DE
* CONCRETO ARMADO COM SEÇÃO QUALQUER SOLICITADOS POR FLEXÃO
* COMPOSTA OBLÍQUA
* DIAG. TENSÃOxDEFORM.DO CONCRETO: POLINÔMIOS DE GRAU QUALQUER
* CONSIDERAÇÃO DA FLUÊNCIA: TEORIA LINEAR DE FLUÊNCIA
* AÇOS BRASILEIROS DO TIPO 'A' OU 'B' COM QUALQUER fy OU Es
* Copyright(c) 1997, Ismael Wilson Cadamuro Júnior
-----
SUBROUTINE INTEGRAL_CONCRETO (N_POLIG,U_POLIG,V_POLIG,e_cg,TETA,
+ H_ALFA,FI_FLUENCIA,m,C,fc,Nrc,Mruc,Mrvc)
INTEGER N_POLIG,m,I,K
REAL*8 U_POLIG(N_POLIG+1),V_POLIG(N_POLIG+1),e_cg,TETA,H_ALFA,

```

```

+ FI_FLUENCIA,C(m+1),fc,Nrc,Mruc,Mrvc,TETA_D,a,b,F0,F1,F2,e_1,e_2,
+ Cf[ALLOCATABLE](:),DELTA_U,DELTA_V,ANGULO
ALLOCATE (Cf(m+1)) !obs: dimensão dos vetores U e V contém
Nrc=0 ! o "+1" para não ocorrer erro no
Mruc=0 ! caso de se entrar com n_polig=0
Mrvc=0
IF (N_POLIG.EQ.0) GOTO 1 !não há poligonal comprimida
IF (N_POLIG.LT.4) THEN !teste de consistência
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) 'ERRO (SUBROTINA INTEGRAL_CONCRETO): POLIGONAL '
+ , 'COMPRIMIDA COM MENOS DE 4 NOS'
STOP
END IF
IF (H_ALFA.EQ.0) THEN !teste de consistência
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) 'ERRO (SUBROTINA INTEGRAL_CONCRETO): ALTURA DA ',
+ 'SECAO IGUAL A ZERO'
STOP
END IF
IF (ABS(TETA).LE.1E-30) THEN !teste de consistência
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) 'ERRO (SUBROTINA INTEGRAL_CONCRETO):TETA ARBITRADO=0'
STOP
END IF
IF (ABS(TETA).LT.1E-6) THEN !teste de aviso de imprecisões
CALL BEEPQQ(1500,1000)
CALL BEEPQQ(1500,1000)
WRITE(*,*) '-----'
WRITE(*,*) 'AVISO (SUBROTINA INTEGRAL_CONCRETO): !TETA! < 1E-6'
WRITE(*,*) ' --> ISTO PODE PROVOCAR IMPRECISOES NOS RESULTADOS',
+ ' DESTA SUBROTINA'
WRITE(*,*) 'Se quiser continuar assim mesmo, tecle <ENTER>.'
WRITE(*,*) '-----'
PAUSE
END IF
TETA_D=TETA/H_ALFA !curvatura majorada (1000*1/r)
K=0
DO WHILE (K.LE.m) !efeito da fluência no diag. tensão-deform. do concr.
Cf(K+1)=C(K+1)/(1+FI_FLUENCIA)**K
K=K+1
END DO
I=1
DO WHILE (I.LE.N_POLIG-1) !loop para cada lado da poligonal
DELTA_U=ABS(U_POLIG(I+1)-U_POLIG(I))
DELTA_V=ABS(V_POLIG(I+1)-V_POLIG(I))
IF (DELTA_U.LE.1E-10.AND.DELTA_V.LE.1E-10) THEN !lado com nós inicial e
I=I+1 !final de mesmas coord.
CYCLE
END IF
ANGULO=ATAN2(DELTA_V,DELTA_U)
IF (ANGULO.LE.1.74E-5.OR.DELTA_V.LE.3E-6) THEN
I=I+1 !lado horizontal (<0.001 grau) não
CYCLE ! contribui com esforços resistentes
END IF ! e causaria erro (divisão por zero)
F0=0 !ou imprecisão numérica do resultado
F1=0
F2=0
e_1=e_CG+TETA_D*V_POLIG(I) !deformação do nó 1 do lado 'i'
e_2=e_CG+TETA_D*V_POLIG(I+1) !deformação do nó 2 do lado 'i'
K=0
DO WHILE (K.LE.m) !cálc.da somatória dos coef. F's
F0=F0+Cf(K+1)/(K+1)*(e_2**(K+1)-e_1**(K+1))
F1=F1+Cf(K+1)/(K+2)*(e_2**(K+2)-e_1**(K+2))
F2=F2+Cf(K+1)/(K+3)*(e_2**(K+3)-e_1**(K+3))
K=K+1
END DO
F0=fc/TETA_D**1*F0 !cálc. dos coef. F's
F1=fc/TETA_D**2*F1
F2=fc/TETA_D**3*F2
a=(U_POLIG(I+1)-U_POLIG(I))/(V_POLIG(I+1)-V_POLIG(I)) !cálc. a e b
b=U_POLIG(I)-a*(V_POLIG(I)+e_CG/TETA_D)

```

```

      Nrc=Nrc+b*F0+a*F1      !soma dos esforços devidos à integral de cada lado
      Mruc=Mruc+b**2*F0/2+b*a*F1+a**2*F2/2
      Mrvc=Mrvc+b*F1+a*F2-e_CG/TETA_D*(b*F0+a*F1)
      I=I+1
END DO
IF (Nrc.LT.0) THEN      !teste de consistência
  CALL BEEPQQ(1500,1000)
  CALL BEEPQQ(1500,1000)
  WRITE(*,*) 'ERRO (SUBROTINA INTEGRAL_CONCRETO): FORCA NORMAL ',
+   'RESISTENTE NEGATIVA --> A POLIGONAL NAO FOI NUMERADA NO ',
+   'SENTIDO TRIGONOMETRICO (ANTI-HORARIO)'
  STOP
END IF
1  DEALLOCATE (Cf)
END
*=====fim da subrotina INTEGRAL_CONCRETO=====

```