

UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS
DEPARTAMENTO DE ESTRUTURAS

CONTRIBUIÇÃO AO ESTUDO DE PILARES EM CONCRETO
ARMADO COM SEÇÃO QUALQUER SUBMETIDOS À
FLEXÃO COMPOSTA OBLÍQUA

ENG. NIVALDO BENEDITO FERREIRA CAMPOS
ORIENTADOR: PROF. DR. WALTER ABRAHÃO NIMIR

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

DEPARTAMENTO DE ESTRUTURAS
ESCOLA DE ENGENHARIA DE SÃO CARLOS

Banca Examinadora:

Prof. Dr. Walter Abrahão Nimir - EESC/USP

Prof. Dr. José Samuel Giongo - EESC/USP

Prof. Dr. Jasson Rodrigues de Figueiredo Filho - UFSCar

SÃO CARLOS, DEZEMBRO DE 1994
SÃO PAULO - BRASIL

À Bernadete, minha esposa
À Natália e ao Francisco, meus filhos
À Amélia e Alberto, meus pais
À Maria Idalina, minha sogra

Dedico este trabalho.

AGRADECIMENTOS

Ao Prof. Dr. Walter Abrahão Nimir, pelo estímulo e orientação recebidos durante a elaboração deste trabalho.

A todos os colegas, professores e funcionários do Departamento de Estruturas, que de alguma forma colaboraram para a realização deste trabalho.

À CAPES, pelo auxílio financeiro.

SUMÁRIO

SUMÁRIO.....	IV
LISTA DE FIGURAS.....	VI
RESUMO.....	VIII
ABSTRACT.....	IX
1. Introdução.....	1.1
1.1. Histórico.....	1.1
1.2. Objetivos do trabalho.....	1.3
2. Caracterização do Concreto Armado.....	2.1
2.1. Generalidades.....	2.1
2.2. Hipóteses Básicas.....	2.1
2.3. Relação Tensão x Deformação para o Concreto.....	2.2
2.4. Relação Tensão x Deformação para o Aço.....	2.4
3. Equações constitutivas dos Materiais.....	3.1
3.1. Estados Limites.....	3.1
3.1.1. Estado Limite Último de Ruína.....	3.2
3.2. Equações de Compatibilidade de Deformações.....	3.5
3.3. Equações de Equilíbrio.....	3.7
4. Flexão Composta Oblíqua.....	4.1
4.1. Caracterização da flexão composta oblíqua no concreto armado.....	4.1
4.2. Problemas Básicos.....	4.1
4.3. Contribuição Resistente do Concreto.....	4.3
4.4. Superfície de Interação.....	4.7
4.5. Dimensionamento Analítico de Seções à Flexão Composta Oblíqua.....	4.9
4.6. Determinação do Estado de Deformações e de Tensões.....	4.16
4.6.1. Processo iterativo para equilibrar as normais atuante e resistente.....	4.17
4.6.2. Processo iterativo para determinação do estado de deformações da seção.....	4.19

4.7. Deformações na Flexão Composta Oblíqua.....	4.25
5. Estudo de Pilares Submetidos à Flexão Composta Oblíqua.....	5.1
5.1. Dimensionamento de Pilares com a consideração da não linearidade física e geométrica.....	5.1
5.2. Solicitações atuantes em estruturas de edifícios.....	5.3
5.3. Excentricidade Acidental.....	5.3
5.4. Consideração da Fluência.....	5.4
6. Programa para Dimensionamento Automático de Pilares Submetidos à Flexão Composta Oblíqua.....	6.1
6.1. Introdução.....	6.1
6.2. Utilização do Programa.....	6.1
6.2.1. Menu Pilares.....	6.3
6.2.2. Menu Seções.....	6.4
6.2.3. Menu Relatórios.....	6.6
6.2.4. Menu Configuração.....	6.7
6.2.5. Entrada de dados.....	6.8
6.2.6. Unidades utilizadas.....	6.8
7. Exemplos de Dimensionamento.....	7.1
8. Conclusões.....	8.1
Bibliografia.....	B.1
Apêndice.....	A.1

LISTA DE FIGURAS

2.1. Diagramas Tensão x Deformação para concretos de diversas resistências.....	2.2
2.2. Diagramas Tensão x Deformação de cálculo do concreto....	2.3
2.3. Diagramas Tensão x Deformação para os aços classes A e B.....	2.5
2.4. Diagramas Tensão x Deformação de cálculo para o aço classe A.....	2.6
2.5. Diagrama convencional de cálculo para os aços classe B..	2.7
3.1. Diagramas últimos de deformação de uma seção de concreto armado.....	3.2
3.2. Domínios de deformação.....	3.3
3.3. Parâmetros da seção transversal utilizados na determinação de sua curvatura.....	3.6
4.1. Seção qualquer de concreto armado formada por contornos poligonais.....	4.4
4.2. Trecho ΔC_k da poligonal C_i , representado pelo segmento k.	4.5
4.3. Superfície de interação de uma seção qualquer de concreto armado.....	4.8
4.4. Processo iterativo para o dimensionamento analítico de seções quaisquer sujeitas à flexão composta oblíqua.....	4.10
4.5. Processo iterativo para a determinação da posição relativa entre P_d e a superfície resistente do concreto.....	4.11
4.6. Diagrama $N \times 1/r$ para uma dada inclinação da LN.....	4.16
4.7. Processo iterativo para determinar o estado de deformações de uma seção qualquer.....	4.20
4.8. Funções f_α e $f(0)$	4.22
4.5. Deformações em uma barra reta sujeita à flexão composta oblíqua.....	4.26

4.6. Excentricidades de primeira ordem e os esforços solicitantes na seção da base da barra.....	4.26
6.1. Tela principal do programa DP.....	6.2
6.2. Sistemas de coordenadas Global e Local.....	6.2
6.3. Seção transversal criada com o programa DP.....	6.6
7.1. Pilar do exemplo 1.....	7.2
7.2. Pilar do exemplo 2.....	7.3
7.3. Pilar do exemplo 3.....	7.4

RESUMO

No dimensionamento de pilares em concreto armado sujeitos a flexão composta oblíqua têm-se utilizado métodos aproximados, os quais muitas vezes não conduzem a resultados satisfatórios ou então faz-se uso de ábacos que, apesar da precisão dos resultados obtidos, não são suficientes para atender a grande diversidade de situações que se apresentam nas estruturas usuais.

Neste trabalho este tema é abordado da forma mais abrangente possível, desenvolvendo-se formulações válidas para barras com seções de forma qualquer, sujeitas à flexão composta oblíqua e utilizando-se o Método Geral para consideração da instabilidade. São apresentados ainda processos iterativos próprios para o uso em computadores, através dos quais é feito o dimensionamento de pilares em concreto armado e com seção qualquer, sendo a não linearidade física e geométrica levada em consideração, não só no dimensionamento como também na determinação dos esforços atuantes.

Aplicam-se estes procedimentos na elaboração de um programa para computador o qual automatiza todas as etapas do dimensionamento de pilares em concreto armado, tornando extremamente simples sua definição, das ações atuantes e suas combinações, bem como a criação de seções transversais que podem ter quaisquer formas e distribuições de armadura.

ABSTRACT

In columns dimensioning in reinforced concrete submitted to a bi-axial bending with axial force, approximated methods have been used, which many times do not lead to satisfactory results, or then abacus can be used that, in spite of the accuracy of the obtained results, are not enough to attend the great diversity of situations that are presented in usual structures.

In this study the theme is broached in a way as embracing as possible, and valid formulations to members with arbitrary cross section have been developed, submitted to a bi-axial bending with axial force and using the General Method for consideration of the instability. Suitable iterative processes are presented for use in computers, through which the dimensioning of columns in reinforced concrete is done, and the physical and geometrical nonlinearity is taken into account, not only in the dimensioning, but also in the determination of the acting efforts.

These proceedings are applied in a program elaboration for computer, which automatizes every dimensioning steps of columns in reinforced concrete, making extremely simple its definition, the characterization of the acting loads and their combinations, as well as the creation of cross sections that can have any shapes and reinforcement distributions.

CAPÍTULO 1

INTRODUÇÃO

1.1. HISTÓRICO

Desde o início da utilização do concreto armado como material estrutural, na segunda metade do século passado, tornou-se imperativo o desenvolvimento de formulações que possibilitassem quantificar a segurança contra a ruptura de estruturas a serem construídas com este material.

Com este objetivo, inúmeros ensaios foram realizados, destacando-se o trabalho de Mörsch, que em 1902 publicou os resultados que obteve e a partir dos quais desenvolveu os conceitos fundamentais da teoria do concreto armado, cujos elementos essenciais são válidos até hoje.

Surgiram assim as primeiras normas técnicas que estabeleceram critérios para a elaboração de projetos estruturais em concreto armado. Desde então houve um contínuo desenvolvimento da tecnologia do concreto e da formulação teórica do comportamento do concreto armado, permitindo que se passasse da utilização de métodos de cálculo baseados na tensão admissível no início do século, para a utilização de métodos semi-probabilísticos nos anos 70.

Nesta mesma época, o advento dos microcomputadores tornou possível a utilização de processos de cálculo bastante refinados, tanto no que diz respeito à determinação dos esforços solicitantes, como no dimensionamento dos elementos estruturais, passando-se a adotar

modelos matemáticos que reproduzem com maior precisão o comportamento real das estruturas, inclusive com a consideração destas serem tridimensionais e, no caso de estruturas em concreto armado, sem linearidade física e geométrica.

Dentre os elementos estruturais, podemos destacar o elemento pilar, o qual devido as suas características, admite solução analítica apenas em alguns casos específicos, necessitando na maioria das vezes, uma abordagem através de métodos iterativos, os quais são viáveis apenas com o uso de computadores. Desta forma, durante muito tempo recorreu-se ao uso de processos aproximados para o dimensionamento de pilares, os quais muitas vezes apresentavam resultados contra a segurança, como foi observado por SAUAIA (1981). Também é corrente a utilização de ábacos para o dimensionamento de pilares, inclusive com a consideração da instabilidade, porém, apesar de precisos, limitam-se aos tipos de pilares para os quais foram elaborados. Esta situação é agravada quando se tem flexão composta oblíqua, na qual, para uma mesma seção, necessitamos de um ábaco para cada força normal e para cada esbeltez considerada.

Esta situação motivou diversos trabalhos, alguns voltados para a automatização da confecção de ábacos, outros para a elaboração de programas que realizassem diretamente o dimensionamento. No entanto, verifica-se que cada uma destas abordagens possui a sua limitação, seja quanto à forma das seções ou dos esforços possíveis de serem considerados, como na adoção de simplificações para a consideração da instabilidade, que dão resultados a favor da segurança, mas contra a economia.

Desta forma, existe a necessidade de se abordar o problema do dimensionamento de estruturas em concreto armado, especificamente no caso de pilares, de uma forma abrangente, aplicável a todas as situações possíveis de serem encontradas no projeto de estruturas em concreto armado. Esta necessidade se torna mais urgente, quando observamos que nas últimas décadas, o desenvolvimento de tecnologias para a produção de concretos de alto desempenho ampliou o

campo de utilização do concreto armado, tornando muitas vezes o seu uso economicamente viável, em países desenvolvidos, como alternativa à estrutura metálica. Porém a melhoria das características do material, nos conduz a concepção de estruturas cada vez mais esbeltas, onde as simplificações usuais nos conduzem a resultados na maioria das vezes distantes da realidade.

1.2. OBJETIVOS DO TRABALHO

Este trabalho tem por objetivo instrumentalizar o engenheiro projetista de estruturas com uma ferramenta a ser utilizada no dimensionamento de pilares em concreto armado. Ao longo dele apresentam-se procedimentos que permitem o dimensionamento de elementos estruturais com seção qualquer, sujeitos à flexão composta oblíqua e com consideração da não linearidade física e geométrica, culminando com a elaboração de um programa para microcomputador que faz uso destes procedimentos e possibilita ao usuário facilmente definir as características dos pilares e de suas seções transversais.

No Capítulo 2 são apresentadas as hipóteses clássicas para o comportamento do concreto armado e as relações Tensão X Deformação do concreto e do aço.

No Capítulo 3 são apresentadas as hipóteses relativas ao Estados Limites Últimos a partir dos quais serão determinadas as deformações últimas e conseqüentemente os esforços resistentes das seções dos elementos a serem dimensionados. Desenvolvem-se também as equações gerais de equilíbrio e de compatibilidade de deformações.

No Capítulo 4 caracteriza-se a flexão oblíqua composta e superfícies de interação, definindo-se procedimentos analíticos para a verificação e dimensionamento de seções.

No Capítulo 5 analisa-se o comportamento de pilares, com a consideração da não linearidade física e geométrica, apresentando-se

procedimentos para a consideração dos efeitos de segunda ordem na determinação dos esforços e no dimensionamento dos elementos estruturais em concreto armado.

No Capítulo 6 descreve-se o programa DP (Dimensionamento de Pilares), o qual foi desenvolvido utilizando-se os procedimentos apresentados nos capítulos anteriores.

Finalmente, no Capítulo 7 apresentam-se exemplos de utilização do programa DP e discute-se os resultados obtidos, comparando-os com os apresentados em outros trabalhos.

CAPÍTULO 2

CARACTERIZAÇÃO DO CONCRETO ARMADO

2.1. GENERALIDADES

Para o estudo de peças em concreto armado, faz-se necessário a caracterização deste material. Desta forma serão adotadas hipóteses relativas ao seu comportamento quando submetido a solicitações normais. Estudar-se-á também os diagramas Tensão x Deformação do concreto e do aço, adotando-se valores convencionais de cálculo. A partir destas considerações pode-se definir, para uma seção qualquer, as equações de equilíbrio de esforços e de compatibilidade de deformações.

2.2. HIPÓTESES BÁSICAS

Nas formulações apresentadas neste trabalho são admitidas as seguintes hipóteses, usuais no estudo de barras de concreto armado submetidas a solicitações normais:

1. **Comportamento Solidário dos Materiais:** O comportamento solidário do concreto e do aço, devido à aderência, garante que a deformação em cada barra seja a mesma do concreto que lhe é adjacente.
2. **Manutenção da Forma Plana da Seção Transversal:** Admite-se, até mesmo no estado limite último, a hipótese de Navier de manutenção da forma plana, ao longo do processo de deformação, das seções

transversais de concreto armado. Decorre desta hipótese uma distribuição plana de deformações normais.

3. A Forma da Seção Transversal não é alterada com as deformações.
4. Não existem Tensões e Deformações iniciais ou residuais, portanto não se tratará de estruturas em concreto protendido.

2.3. RELAÇÃO TENSÃO X DEFORMAÇÃO PARA O CONCRETO

O concreto é um material elásto-plástico, apresentando, porém, um comportamento aproximadamente elástico-linear para tensões até 30% de sua máxima tensão de compressão. Quando este valor é ultrapassado, inicia-se a plastificação, dando origem a um diagrama tensão x deformação curvo.

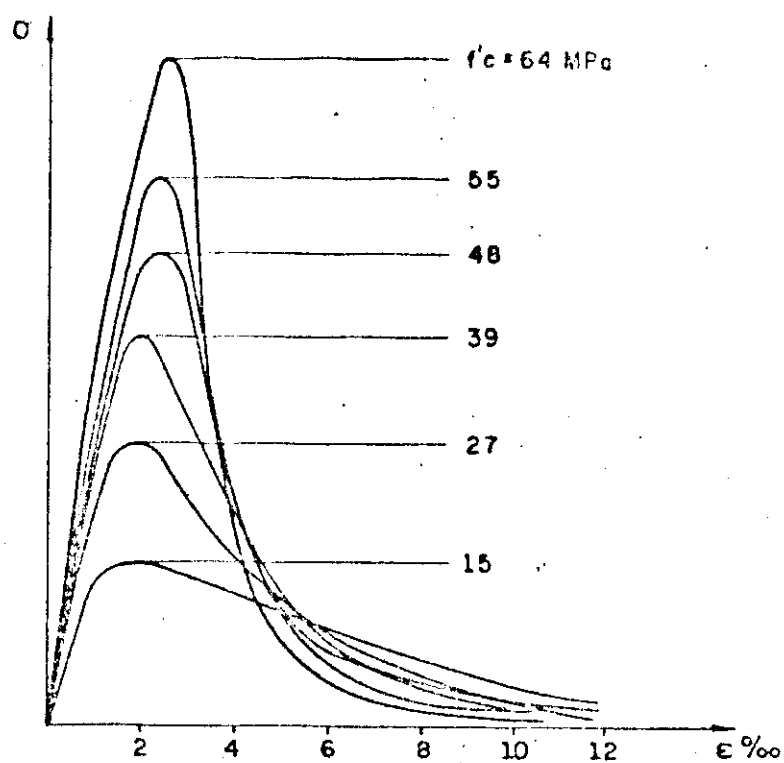


Fig. 2.1 - Diagramas tensão x deformação para concretos de diversas resistências, ensaiados à compressão axial. (PINHEIRO, 1986)

Observa-se pela figura 2.1, que o pico de tensões de compressão no concreto acontece aproximadamente para uma deformação igual a 2‰ independentemente do concreto utilizado. No entanto, estes resultados refletem condições particulares de ensaio, pois dependem de uma série de fatores, tais como: velocidade do carregamento, forma e dimensões do corpo-de-prova, tipo de carregamento aplicado, etc.

Desta forma, deve-se escolher um diagrama de cálculo que apresente a melhor concordância possível com os dados experimentais, garantindo a segurança suficiente para cobrir as divergências entre as condições em que o concreto foi ensaiado e as de utilização.

A NBR-6118 adota o diagrama simplificado parábola-retângulo (fig. 2.2) para o dimensionamento no estado limite último de peças em concreto armado com qualquer forma de seção transversal. Neste trabalho considera-se as tensões de tração e as deformações devidas a elas com valores negativos.

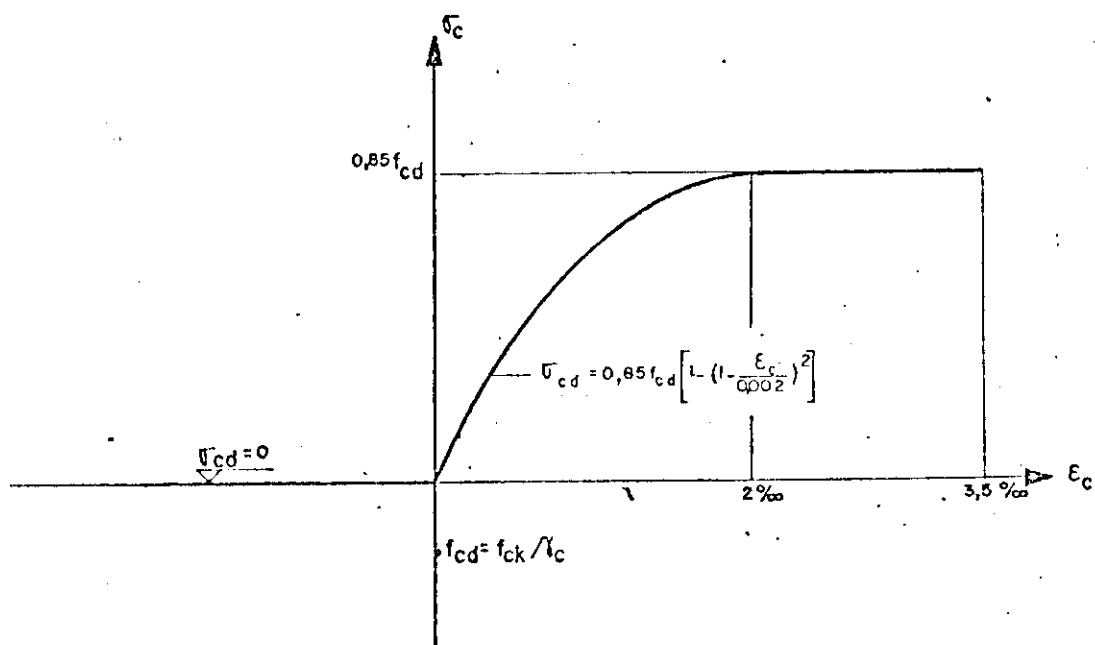


Fig. 2.2 - Diagrama tensão x deformação de cálculo do concreto (FORNI, 1980)

A curva tensão x deformação é composta de uma parábola do 2º grau em ϵ_c que passa pela origem e tem seu vértice no ponto de

abscissa 2‰ e ordenada $0,85 f_{cd}$ e de uma reta entre as deformações 2‰ e 3,5‰, tangente à parábola e paralela ao eixo das abscissas.

Ela é descrita pelas equações:

$$\sigma_c = 0 \quad \text{para } \epsilon_c \leq 0 \quad (2.1)$$

$$\sigma_c = 0,85f_{cd} [1 - [1 - \epsilon_c/0,002]^2] \quad \text{para } 0 \leq \epsilon_c \leq 2‰ \quad (2.2)$$

$$\sigma_c = 0,85 f_{cd} \quad \text{para } 2‰ \leq \epsilon_c \leq 3,5‰ \quad (2.3)$$

A tensão máxima de cálculo σ_{cd} é função da resistência de cálculo f_{cd} , sendo esta determinada a partir do valor de f_{ck} , resistência característica do concreto à compressão, e γ_c , coeficiente de minoração da resistência, o qual vale geralmente 1,4, de acordo com o item 5.4.1 da NBR-6118.

Este coeficiente deve ser tomado igual a 1,5 quando se tiver condições desfavoráveis de concretagem e igual a 1,3 no caso de peças pré-moldadas em usina, executadas com cuidados rigorosos. Assim temos que:

$$f_{cd} = f_{ck} / \gamma_c \quad (2.4)$$

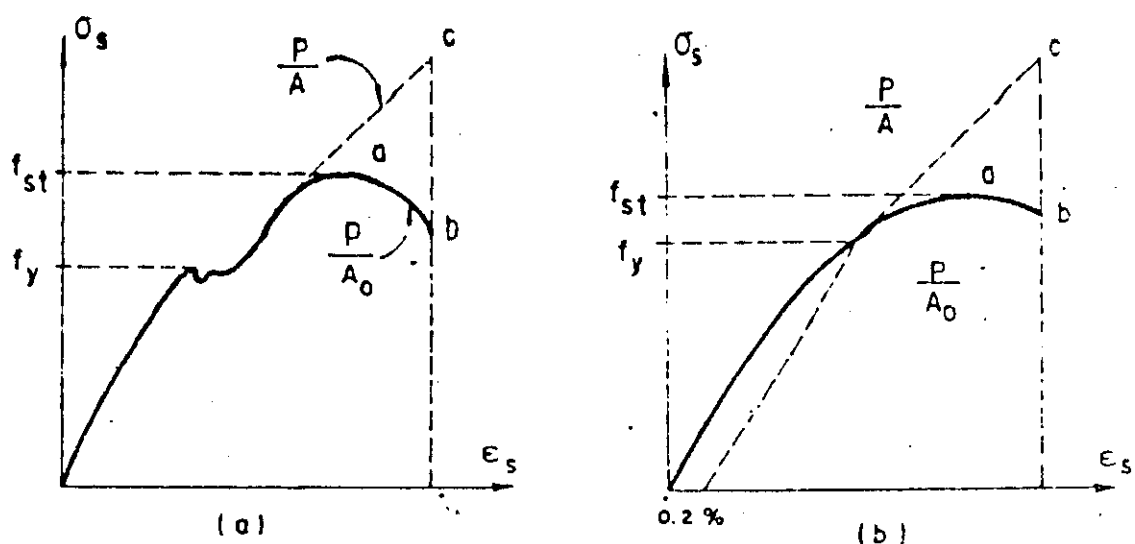
$$\sigma_{cd} = 0,85 f_{cd} \quad (2.5)$$

2.4. RELAÇÃO TENSÃO X DEFORMAÇÃO PARA O AÇO

Os aços utilizados na construção civil para a obtenção de concreto armado são os aços-carbono. Eles se classificam em aços de dureza natural ou aços classe A e aços encruados a frio ou classe B.

Os aços classe A, laminados a quente e resfriados ao ar livre, caracterizam-se por apresentarem um diagrama tensão x deformação linear até o escoamento, a partir do qual forma-se um patamar de escoamento bem definido (fig. 2.3a).

Já os aços classe B, encruados a frio por tração, torção ou compressão, não apresentam patamar no diagrama tensão x deformação (fig. 2.3.b).



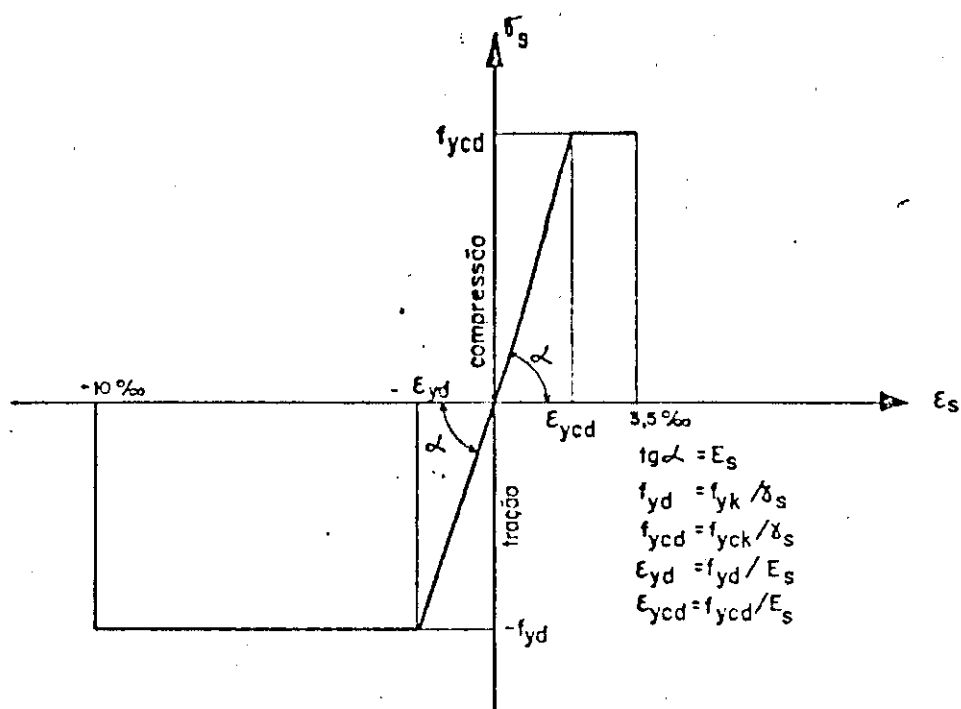
- P - força aplicada
- A - área da seção em cada instante
- A_0 - área inicial da seção
- a - é o ponto da curva, correspondente à resistência convencional à ruptura
- b - é o ponto da curva, correspondente à resistência aparente à ruptura
- c - é o ponto da curva, correspondente à resistência real à ruptura

Fig. 2.3 - Diagramas tensão x deformação para os aços classes A e B (PINHEIRO, 1986)

A NBR-6118 adota um diagrama de cálculo simplificado para o aço classe A (fig 2.4), admitindo ser este um material elasto-plástico perfeito.

Os limites de deformação são fixados em 10‰ para o alongamento específico máximo e em 3,5‰ para o encurtamento específico máximo. Tanto para o aço classe A como para o classe B podemos adotar para o módulo de deformação longitudinal E_s um valor constante e igual a 21000 kN/cm². Para o coeficiente de minoração da resistência do aço, γ_s , adota-se o valor igual a 1,15.

Os valores das resistências características do aço na compressão, f_{yck} e na tração, f_{yk} , podem, na falta de dados experimentais, ser considerados iguais.



f_{y_k} - resistência característica do aço à tração

$f_{y_{ck}}$ - resistência característica do aço à compressão

γ_s - coeficiente de minoração da resistência do aço

f_{y_d} - resistência de cálculo do aço à tração

$f_{y_{cd}}$ - resistência de cálculo do aço à compressão

E_{y_d} - deformação específica de cálculo de início de escoamento do aço na tração

$E_{y_{cd}}$ - deformação específica de cálculo de início de escoamento do aço na compressão

E_s - módulo de deformação longitudinal do aço

Fig. 2.4- Diagrama tensão x deformação de cálculo para o aço classe A (FORNI, 1980).

Para o aço classe B a NBR-6118 adota um diagrama tensão x deformação convencional de cálculo (fig. 2.5) caracterizado por três trechos:

TRECHO 1 - Linear para $0 \leq \sigma_s \leq 0,7 f_{yd}$ (2.6)

TRECHO 2 - Parabólico entre o ponto de tensão $\sigma_s = 0,7 f_{yd}$ e o ponto relativo à tensão de escoamento convencional. Este trecho é definido pelas equações 2.7 ou 2.8.

$$\epsilon_s = \frac{\sigma_s}{E_s} + \frac{1}{45} \left(\left| \frac{\sigma_s}{f_{yd}} \right| - 0,7 \right)^2 \cdot \left(\frac{\sigma_s}{f_{yd}} \right) \quad (2.7)$$

$$\sigma_s = f_{yd} \left(0,70 - \frac{45 f_{yd}}{2 E_s} + \sqrt{\left(\frac{45 f_{yd}}{2 E_s} - 0,70 \right)^2 + \left(45 \left| \epsilon_s \right| - 0,49 \right)} \right) \quad (2.8)$$

TRECHO 3- Patamar convencional de escoamento, com tensão $\sigma_s = f_{yd}$. O início do patamar de escoamento corresponde a uma deformação específica dada por:

$$\epsilon_{yd} = 2\text{‰} + f_{yd}/E_s \quad (2.9)$$

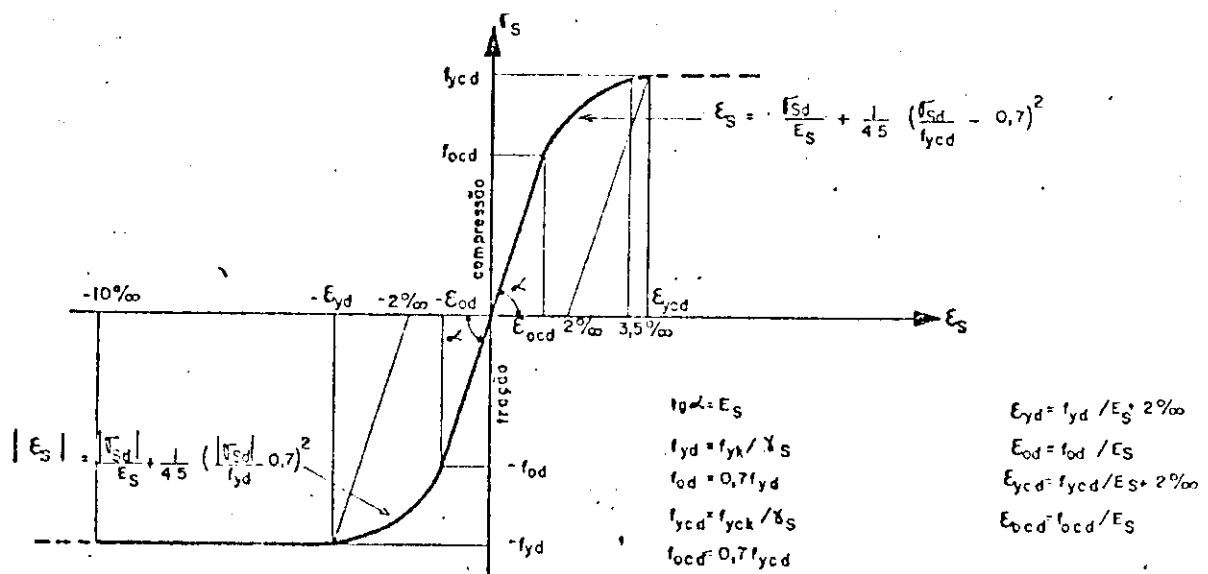


fig. 2.5 - Diagrama convencional de cálculo para os aços classe B (FORNI, 1980).

CAPÍTULO 3

EQUAÇÕES CONSTITUTIVAS DOS MATERIAIS

3.1 - ESTADOS LIMITES

Denomina-se de modo convencional, segundo o CEB/78, Estado Limite como o estado em que uma estrutura, total ou parcialmente, não possa ser considerada utilizável para o propósito para o qual ela foi concebida, seja por ter sido ultrapassada a sua capacidade resistente, como por não atender as condições de serviço requeridas. Pode-se agrupar os estados limites em duas categorias:

a) estados limites últimos, caracterizados pelo esgotamento da capacidade portante da estrutura. Neste estado incluem-se:

- perda da estabilidade de uma parte ou toda a estrutura;
- ruptura das seções críticas da estrutura;
- transformação da estrutura num mecanismo;
- deterioração por efeito de fadiga;
- deformação plástica excessiva.

b) estados limites de utilização, caracterizados pelos diferentes critérios de desempenho para assegurar a utilização normal e durabilidade de uma estrutura. Neles estão incluídos:

- deformações excessivas para o uso normal da estrutura;
- fissuração prematura ou excessiva;
- corrosão;
- deslocamentos excessivos, sem perda do equilíbrio;
- vibrações excessivas.

Neste trabalho não é abordado o estado limite de utilização, tratando-se apenas das situações em que temos estado limite último com a ruína da seção.

3.1.1. ESTADO LIMITE ÚLTIMO DE RUÍNA

A caracterização do estado limite último de ruptura ou de deformação plástica excessiva será feita de forma convencional, devido à dificuldade de se estabelecer com precisão o esgotamento da capacidade resistente das peças submetidas a solicitações normais. A figura 3.1 apresenta os diagramas de deformações últimas para uma seção de concreto armado.

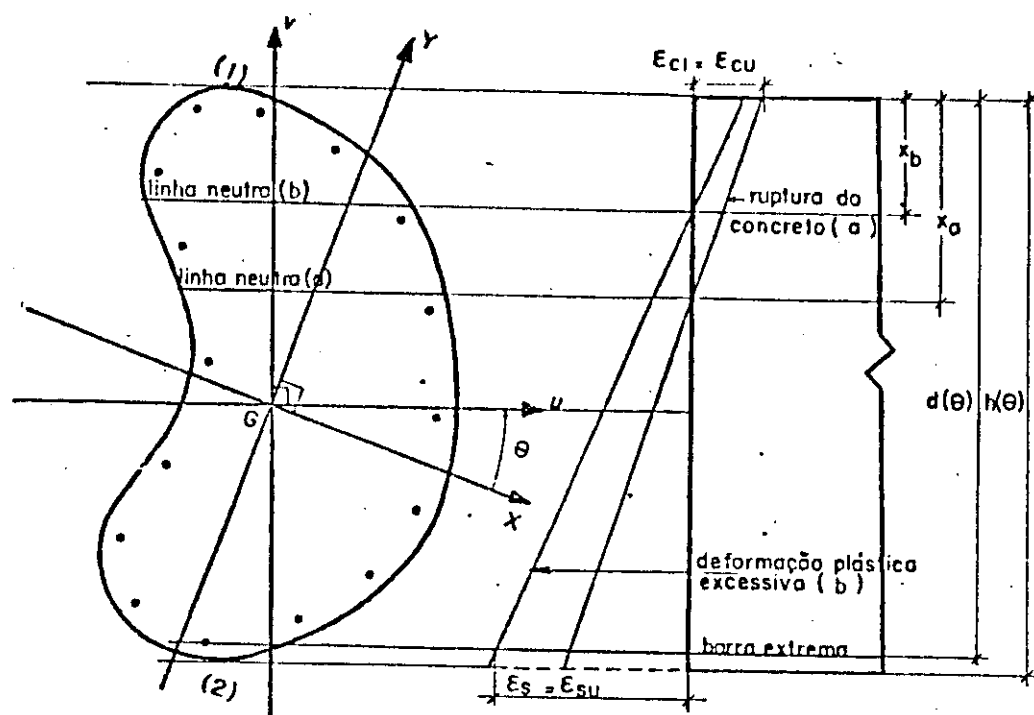


Fig. 3.1 - Diagramas de deformações últimas de uma seção de concreto armado, para uma determinada inclinação da linha neutra (FORNI, 1980).

A NBR 6118, baseada em dados obtidos experimentalmente e apresentados no capítulo 2 deste trabalho, fixa os valores convencionais para o encurtamento máximo do concreto comprimido e para o alongamento máximo da armadura tracionada. A limitação referente à armadura tem por objetivo impedir o surgimento de fissuras com abertura excessiva no concreto tracionado.

Ficam, portanto, definidas três situações que conduzem a seção a um Estado Limite Último:

- alongamento máximo da armadura igual a $-0,01$
- encurtamento máximo do concreto igual a $0,0035$ em seções não inteiramente comprimidas
- encurtamento do concreto, em seções inteiramente comprimidas, não superior a $0,002$ na fibra situada a $3h/7$ da fibra mais encurtada.

Cada uma destas situações define um pólo de deformação, a partir dos quais ficam caracterizados os Domínios de deformação, apresentados na figura 3.2, a qual também sintetiza todas as possíveis configurações últimas do diagrama de deformações específicas, ao longo de uma seção transversal.

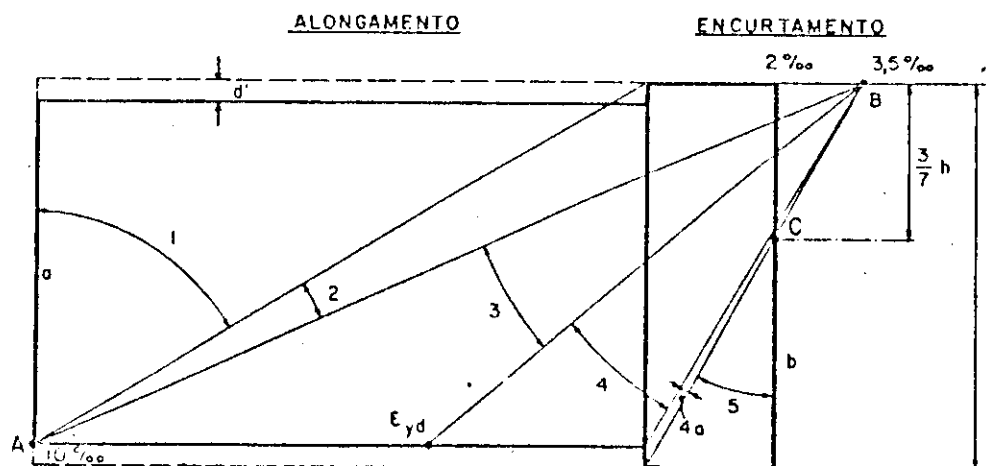


Fig. 3.2 - Domínios de deformação (FORNI, 1980)

RETA a - Tração uniforme. O estado limite último é atingido por deformação plástica excessiva da armadura. Toda a seção está sujeita a um mesmo alongamento;

DOMÍNIO 1 - Tração não uniforme. Toda a seção está tracionada. O estado limite último é atingido por deformação plástica excessiva da armadura;

DOMÍNIO 2 - Flexão simples ou composta sem ruptura do concreto a compressão e com máximo alongamento permitido na armadura;

DOMÍNIO 3 - Flexão simples ou composta com ruptura do concreto a compressão e escoamento da armadura;

DOMÍNIO 4 - Flexão simples ou composta com ruptura do concreto por compressão e aço tracionado sem atingir a tensão de escoamento;

DOMÍNIO 4a - Flexão composta com armaduras comprimidas e pequena região do concreto tracionada;

DOMÍNIO 5 - Compressão não uniforme com toda a seção de concreto comprimida;

RETA b - Compressão uniforme. Toda a seção está sujeita a um mesmo encurtamento.

A partir dos pólos de deformação pode-se englobar todos os domínios de deformação em três regiões:

REGIÃO 1: reta a , domínios 1 e 2;

REGIÃO 2: domínios 3, 4 e 4a;

REGIÃO 3: domínio 5 e reta b.

Temos assim definidas todas as situações em que é atingido o estado limite último por ruptura do concreto ou por alongamento excessivo da armadura.

3.2. EQUAÇÕES DE COMPATIBILIDADE DE DEFORMAÇÕES

Em decorrência das hipóteses adotadas para o comportamento do concreto armado, pode-se admitir a distribuição plana de deformações ao longo da seção transversal. Desta característica resulta que a deformação em qualquer ponto desta seção pode ser determinada através do conhecimento de apenas três parâmetros:

- a- inclinação θ da linha neutra, sendo esta definida como o lugar geométrico dos pontos com deformação nula. O ângulo θ é medido no sentido anti-horário entre o eixo X e a linha neutra;
- b- deformação ϵ_{c1} , correspondente ao ponto de máximo encurtamento ou mínimo alongamento da seção;
- c- curvatura da seção, na direção perpendicular à linha neutra.

A curvatura da seção é dada por:

$$\frac{1}{r(\theta)} = \frac{\epsilon_{c1} - \epsilon_s}{d(\theta)} \quad (3.1)$$

Onde: $d(\theta)$ = distância entre a fibra mais comprimida ou menos alongada e a armadura mais alongada ou menos encurtada

$1/r(\theta)$ = curvatura da seção na direção perpendicular à linha neutra

ϵ_s = deformação específica da armadura com máximo alongamento ou mínimo encurtamento.

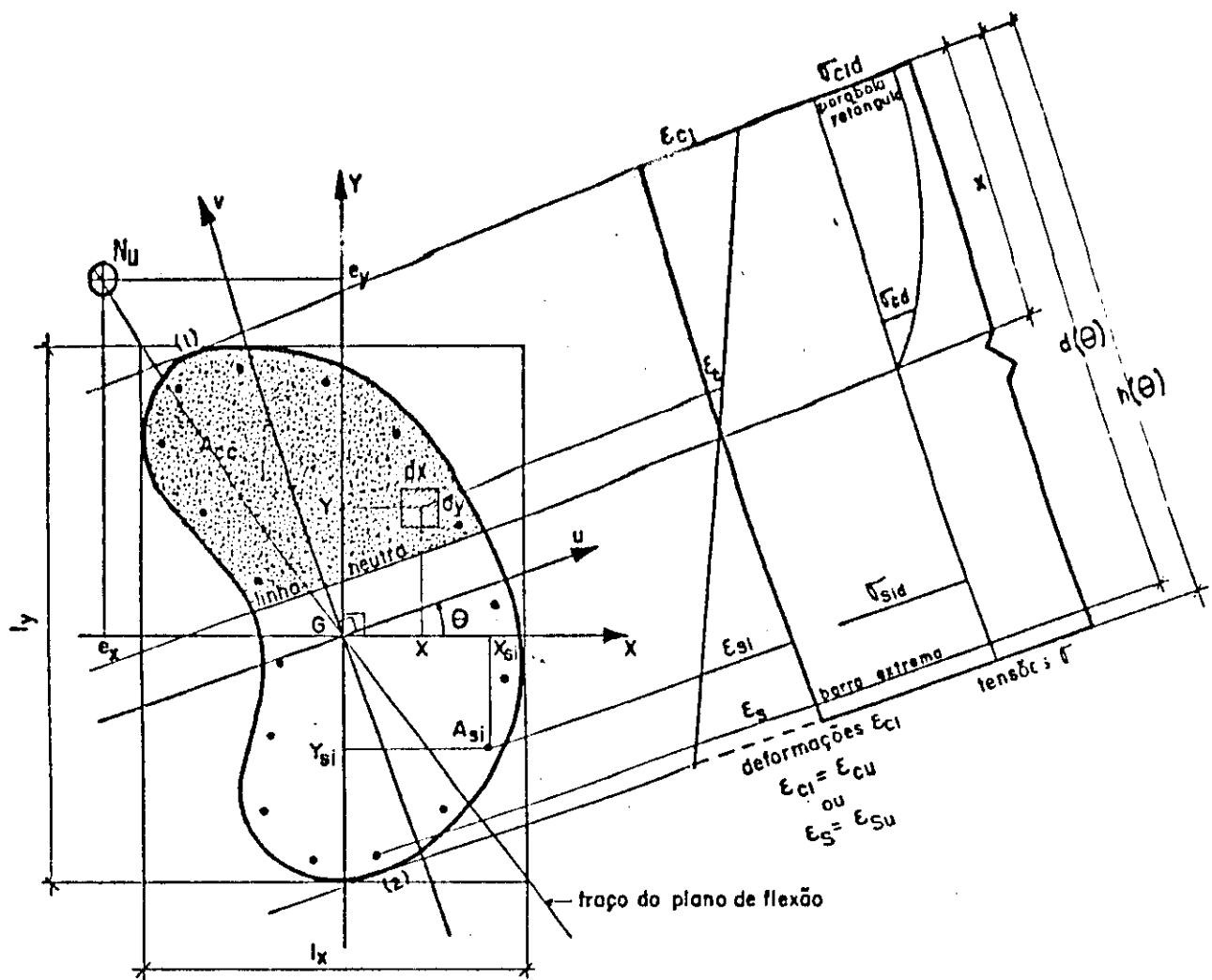


Fig. 3.3 - Parâmetros da seção transversal utilizados na determinação de sua curvatura (FORNI, 1980).

É demonstrado por Fusco (1980) que:

$$\frac{1}{r_x} = \frac{1}{r_\theta} \cdot \cos\theta \quad (3.2)$$

$$\frac{1}{r_y} = \frac{1}{r_\theta} \cdot \sin\theta \quad (3.3)$$

Definindo-se

$$\beta_v = v/d(\theta) \quad (3.4)$$

$$\beta_{v\max} = v_{\max}/d(\theta) \quad (3.5)$$

$$\beta_x = x/d(\theta) \quad (3.6)$$

Temos a partir da fig. 3.3. que a deformação específica em uma fibra qualquer de ordenada v é dada por:

$$\epsilon = (v - v_{\text{máx}} + x) \cdot 1/r \quad (3.7)$$

Onde $v_{\text{máx}}$ é a ordenada da fibra mais comprimida ou menos tracionada da seção e x é a distância entre $v_{\text{máx}}$ e v_{LN} (ordenada v da Linha Neutra).

De 3.4, 3.5 e 3.6 temos:

$$\epsilon = d(\theta) \cdot (\beta_v - \beta_{v_{\text{máx}}} + \beta_x) \cdot 1/r \quad (3.8)$$

Com relação ao parâmetro β_x podemos, em função da curvatura da seção, determinar os valores mínimo e máximo que ele pode assumir. Assim, de Paula (1989), temos:

Para o valor mínimo ou inferior:

$$\beta_{xi} = 1 - 0,01 / (d(\theta)/r) \quad (\text{Região 1}) \quad (3.9)$$

Para o valor máximo ou superior:

$$\beta_{xs} = 0,0035 / (d(\theta)/r) \quad (\text{Região 2}) \quad (3.10)$$

$$\beta_{xs} = 0,002 / (d(\theta)/r) + (3/7) / (h(\theta)/d(\theta)) \quad (\text{Região 3}) \quad (3.11)$$

3.3. EQUAÇÕES DE EQUILÍBRIO

Considerando-se uma seção qualquer de concreto armado, submetida a flexão composta oblíqua com normal N e momento fletor dado por suas componentes M_x e M_y , nela surgem deformações e tensões normais. Estas tensões equilibram os esforços solicitantes de modo a garantir o equilíbrio da seção. Assim, para o estado limite último, pode-se escrever as equações de equilíbrio como a seguir:

$$N_u = \iint_{Acc} \sigma_{cd} dx dy + \sum_{i=1}^n A_{si} \sigma_{sid} \quad (3.12)$$

$$M_{xu} = - \left(\iint_{Acc} \sigma_{cd} y dx dy + \sum_{i=1}^n A_{si} \sigma_{sid} y_{si} \right) \quad (3.13)$$

$$M_{yu} = \iint_{Acc} \sigma_{cd} x dx dy + \sum_{i=1}^n A_{si} \sigma_{sid} x_{si} \quad (3.14)$$

Onde: N_u = normal última

M_{xu} = componente X do momento fletor último

M_{yu} = componente Y do momento fletor último

A_{cc} = área comprimida de concreto

σ_{cd} = tensão de cálculo numa fibra de concreto comprimida, de coordenadas X e Y

A_{si} = área da barra de aço genérica i, de coordenadas X_{si} e Y_{si}

σ_{sid} = tensão de cálculo na barra i

n = número de barras de aço da armadura

CAPÍTULO 4

FLEXÃO COMPOSTA OBLÍQUA

4.1. CARACTERIZAÇÃO DA FLEXÃO COMPOSTA OBLÍQUA NO CONCRETO ARMADO

Entende-se por flexão composta oblíqua (FCO) a solicitação normal, composta de momentos fletores e força normal, que atua em uma seção, sem que se conheça a priori a direção da linha neutra. No estudo da FCO no concreto estrutural, toma-se como pólo de redução dos esforços solicitantes o centro de gravidade da peça, sem consideração da armadura (FORNI, 1980).

A FCO no concreto armado ocorre:

- em seções que por sua forma, não apresentam nenhum plano de simetria
- em seções que, sendo simétricas quanto à forma, são armadas assimetricamente em relação ao plano de simetria
- em seções que, sendo simétricas quanto à forma e disposição da armadura, estão submetidas a solicitações fora do plano de simetria.

4.2. PROBLEMAS BÁSICOS

Para uma dada seção de concreto armado, sendo conhecidas sua forma, a posição e a área de cada barra constituinte de sua armadura, colocam-se dois problemas básicos:

1. Sendo dados o terno de esforços solicitantes N , M_x , M_y , que não ultrapassam a capacidade resistente da seção, qual é o correspondente estado de deformações, e conseqüentemente o estado de tensões, caracterizado pela posição da linha neutra e por sua curvatura, ou seja pelo terno $(\theta, \beta_x, 1/r)$?
2. Assumindo-se um estado de deformações compatível com os limites de resistência da seção, pela fixação da linha neutra e sua curvatura, isto é, fixando-se o terno $(\theta, \beta_x, 1/r)$, quais são os esforços solicitantes (N, M_x, M_y) que equilibram a seção?

A solução do segundo problema pode ser obtida diretamente da aplicação das equações de equilíbrio e de compatibilidade e das relações tensão x deformação do concreto e do aço. A partir do estado de deformações definido pelo terno $(\theta, \beta_x, 1/r)$, obtem-se a deformação ϵ_c em qualquer fibra do concreto e a deformação ϵ_{sl} para qualquer barra da armadura. Com estes valores nos diagramas tensão x deformação do aço e do concreto obtem-se o estado de tensões na seção. Definidas as tensões, as equações de equilíbrio 3.12, 3.13 e 3.14 fornecem, respectivamente, N , M_x , M_y , os esforços solicitantes que equilibram a seção.

Já a solução do primeiro problema não é direta, pois a partir de N , M_x , M_y e utilizando-se as equações de equilíbrio e compatibilidade e as relações tensão x deformação do aço e do concreto, não se consegue determinar de uma forma direta o estado de deformações ou de tensões. Pode-se chegar à solução utilizando-se iterativamente o mesmo procedimento empregado na solução do segundo problema. Adotando-se um valor inicial para θ , $\beta_x, 1/r$, determinam-se os valores correspondentes de N , M_x , M_y . Caso estes não coincidam a menos de uma tolerância pré-definida com os esforços solicitantes, adotam-se novos valores para θ , $\beta_x, 1/r$, e repete-se o procedimento até que a tolerância seja atingida.

4.3. CONTRIBUIÇÃO RESISTENTE DO CONCRETO

Para que se possa aplicar os procedimentos apresentados no item 4.2, é fundamental o conhecimento da contribuição resistente do concreto. Isto é possível através das equações 3.12, 3.13 e 3.14, nas quais se conhece a distribuição retangular-parabólica de tensões na porção da seção de concreto sujeita a compressão (A_{cc}).

A resolução destas equações é obtida transformando-se as integrais de superfície que se estendem sobre a área da seção, em integrais curvilíneas, estendidas ao seu contorno, suposto poligonal, através da aplicação do teorema de Green, a qual é detalhadamente apresentada por Forni (1980).

Na figura 4.1 é apresentada uma seção qualquer de concreto armado formada por contornos poligonais com distribuição parabólica-retangular de tensões no concreto. Observa-se que para a utilização do teorema de Green, a circuitação da poligonal externa deve ser feita no sentido anti-horário e para as poligonais internas no sentido horário.

Assim obtem-se o valor dos esforços resistentes do concreto através das seguintes integrais curvilíneas:

$$N_{uc} = \oint_{C_1} \sigma_{cd} u dv + \sum_{i=2}^{n+1} \oint_{C_{n+1}} \sigma_{cd} u dv \quad (4.1)$$

$$M_{uuc} = - \left(\oint_{C_1} \sigma_{cd} u v dv + \sum_{i=2}^{n+1} \oint_{C_i} \sigma_{cd} u v dv \right) \quad (4.2)$$

$$M_{vuc} = \oint_{C_1} \frac{\sigma_{cd} u^2}{2} dv + \sum_{i=2}^{n+1} \oint_{C_i} \frac{\sigma_{cd} u^2}{2} dv \quad (4.3)$$

Onde C_i representa cada um dos contornos poligonais da seção.

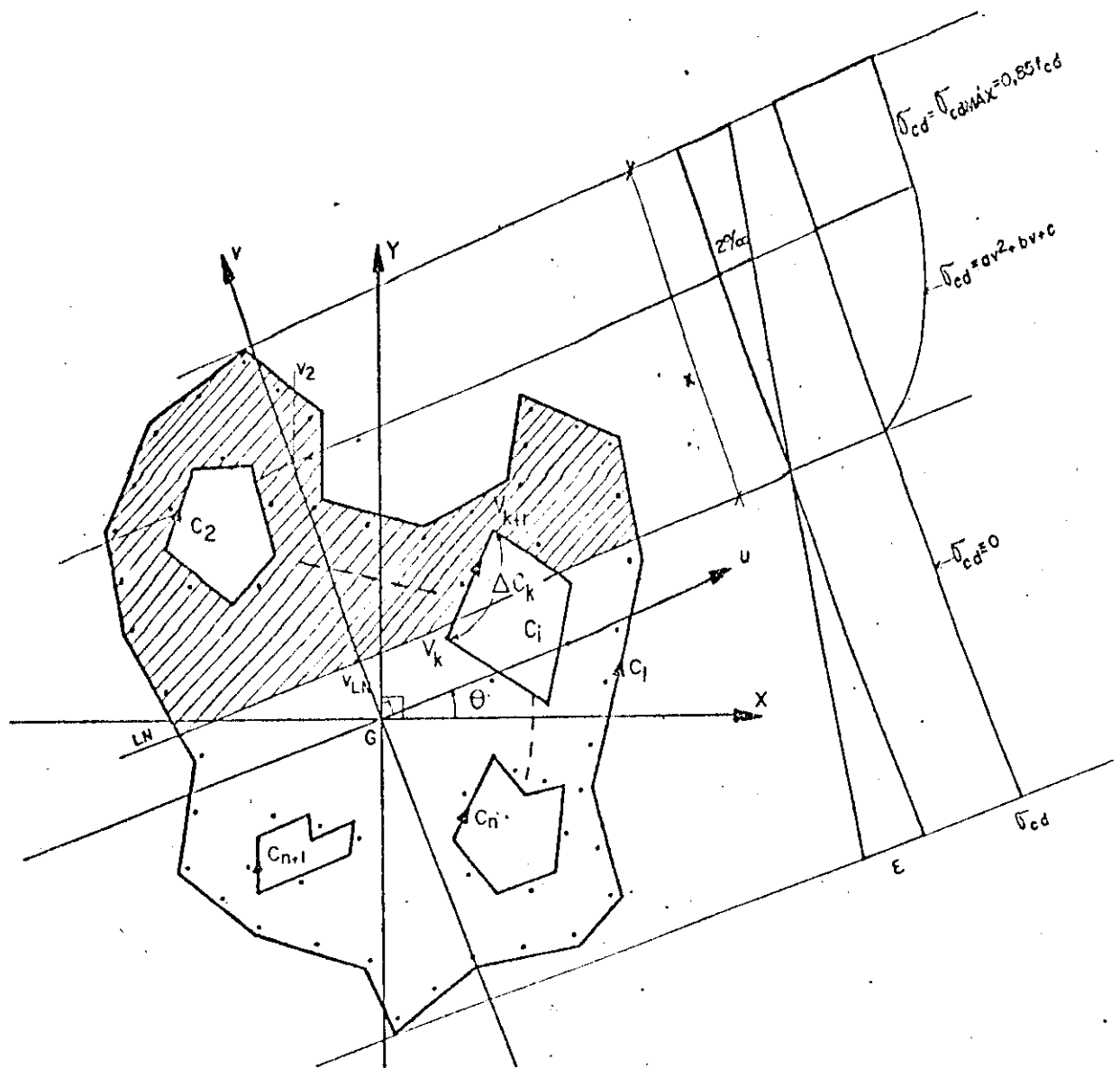


Fig. 4.1- Seção qualquer de concreto armado formada por contornos poligonais. (FORNI, 1980)

Para um dado contorno pode-se colocar a integral curvilínea como a somatória das integrais curvilíneas ao longo de cada segmento que o compoem.

$$\oint_{C_i} \sigma_{cd} u dv = \sum_{k=1}^{n_i} \int_{\Delta C_k} \sigma_{cd} u dv \quad (4.4)$$

$$\oint_{C_i} \sigma_{cd} u v dv = \sum_{k=1}^{n_i} \int_{\Delta C_k} \sigma_{cd} u v dv \quad (4.5)$$

$$\oint_{C_i} \frac{\sigma_{cd} u^2}{2} dv = \sum_{k=1}^{n_i} \int_{\Delta C_k} \frac{\sigma_{cd} u^2}{2} dv \quad (4.6)$$

As tensões em função da ordenada v para cada fibra do concreto são dadas por:

$$v \leq v_{LN} \Rightarrow \sigma_{cd} = 0 \quad (4.7)$$

$$v_{LN} \leq v \leq v_2 \Rightarrow \sigma_{cd} = a v^2 + b v + c \quad (4.8)$$

onde $a = -\sigma_{cd,m\acute{a}x} / (v_2 - v_{LN})^2$

$$b = -2 v_2 a$$

$$c = v_{LN}(2 v_2 - v_{LN}) a$$

$v_2 =$ ordenada v das fibras com encurtamento $20/100$

$$v \geq v_2 \Rightarrow \sigma_{cd} = \sigma_{cd,m\acute{a}x} = 0,85 f_{cd} \quad (4.9)$$

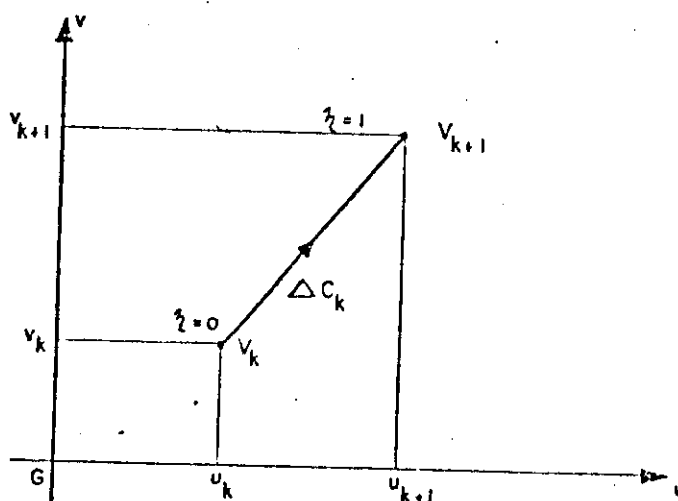


Fig. 4.2 - Trecho ΔC_k da poligonal C_i , representado pelo segmento k (FORNI, 1980)

A partir da fig. 4.2 obtém-se a representação paramétrica do trecho ΔC_k , definido pelos vértices V_k e V_{k+1} e pertencente ao contorno C_i :

$$u = u_k + \xi l_u \quad (4.10)$$

$$\Delta C_k \Rightarrow v = v_k + \xi l_v \quad (4.11)$$

Onde $l_u = u_{k+1} - u_k \quad (4.12)$

$$l_v = v_{k+1} - v_k \quad (4.13)$$

$$0 \leq \xi \leq 1$$

Substituindo 4.10 e 4.11 em 4.7, 4.8 e 4.9, obtemos a distribuição de tensões em função de ξ :

$$v \leq v_{LN} \quad \Rightarrow \quad \sigma_{cd}(\xi) = 0 \quad (4.14)$$

$$v_{LN} \leq v \leq v_2 \quad \Rightarrow \quad \sigma_{cd}(\xi) = a_1 \xi^2 + b_1 \xi + c_1 \quad (4.15)$$

onde

$$a_1 = (v_{k+1} - v_k)^2 a$$

$$b_1 = -2 (v_k - v_2) (v_{k+1} - v_k) a$$

$$c_1 = (v_k - v_{LN}) (v_k + v_{LN} - 2v_2) a$$

$v_2 =$ ordenada v das fibras com encurtamento 2‰

$$a = -\sigma_{cd,m\acute{a}x} / (v_2 - v_{LN})^2$$

$$v \geq v_2 \quad \Rightarrow \quad \sigma_{cd}(\xi) = \sigma_{cd,m\acute{a}x} = 0,85 f_{cd} \quad (4.16)$$

Aplicando-se as equações 4.4, 4.5 e 4.6 ao segmento g\nerico ΔC_k , e assumindo a distribui\c{a}o de tens\~oes definida em 4.14, 4.15 e 4.16, obtem-se:

a) Se o segmento encontrar-se totalmente na regi\~ao tracionada:

$$\int_{\Delta C_k} \sigma_{cd} u dv = 0 \quad (4.17)$$

$$\int_{\Delta C_k} \sigma_{cd} u v dv = 0 \quad (4.18)$$

$$\int_{\Delta C_k} \frac{\sigma_{cd} u^2}{2} dv = 0 \quad (4.19)$$

b) Se o segmento encontrar-se totalmente na regi\~ao com tens\~oes constantes:

$$\int_{\Delta C_k} \sigma_{cd} u dv = \frac{\sigma_{cd,m\acute{a}x}}{2} (v_{k+1} - v_k) (u_k + u_{k+1}) \quad (4.20)$$

$$\int_{\Delta C_k} \sigma_{cd} u v dv = \frac{\sigma_{cd,m\acute{a}x}}{6} (v_{k+1} - v_k) (2u_k v_k + u_k v_{k+1} + u_{k+1} v_k + 2u_{k+1} v_{k+1}) \quad (4.21)$$

$$\int_{\Delta C_k} \sigma_{cd} \frac{u^2}{2} dv = \frac{\sigma_{cd,m\acute{a}x}}{6} (v_{k+1} - v_k) ((u_k + u_{k+1})^2 - u_k u_{k+1}) \quad (4.22)$$

c) Se o segmento encontrar-se totalmente na região com distribuição parabólica de tensões:

$$\int_{\Delta C_k} \sigma_{cd} u dv = \frac{(v_{k+1} - v_k)}{12} (u_{k+1} (3a_1 + 4b_1 + 6c_1) + u_k (a_1 + 2b_1 + 6c_1)) \quad (4.23)$$

$$\int_{\Delta C_k} \sigma_{cd} uv dv = \left(\frac{v_{k+1} - v_k}{60} \right) (u_k v_k (2a_1 + 5b_1 + 20c_1) + (u_k v_{k+1} + u_{k+1} v_k) \cdot (3a_1 + 5b_1 + 10c_1) + u_{k+1} v_{k+1} (12a_1 + 15b_1 + 20c_1)) \quad (4.24)$$

$$\int_{\Delta C_k} \frac{\sigma_{cd} u^2}{2} dv = \left(\frac{v_{k+1} - v_k}{120} \right) (u_k^2 (2a_1 + 5b_1 + 20c_1) + 2u_k u_{k+1} (3a_1 + 5b_1 + 10c_1) + u_{k+1}^2 (12a_1 + 15b_1 + 20c_1)) \quad (4.25)$$

d) Se o segmento não se encontrar totalmente em uma das regiões anteriores, deverá ser dividido em novos seguimentos nos quais isso acontece, aplicando-se a cada um deles as equações adequadas à região em que ele se encontra.

Repetindo-se o cálculo para todos os segmentos do contorno C_i e somando-se os resultados, obtemos as integrais ao longo de todo o contorno. Calculando-se estas integrais para todos os contornos e somando-se os resultados determinam-se os esforços resistentes últimos N_{uc} , M_{uuc} e M_{vuc} .

Calculadas as contribuições resistentes do concreto nos eixos u e v , pode-se exprimi-las em função dos eixos X e Y através de:

$$M_{xuc} = M_{uuc} \cos \theta - M_{vuc} \sin \theta \quad (4.26)$$

$$M_{yuc} = M_{uuc} \sin \theta + M_{vuc} \cos \theta \quad (4.27)$$

4.4. SUPERFÍCIE DE INTERAÇÃO

Para uma dada seção de concreto armado, pode-se caracterizar sua resistência a solicitações normais pelo conjunto de todas as solicitações últimas (N_u, M_{xu}, M_{yu}) que a esgotam. A representação destas solicitações últimas no espaço $ONM_x M_y$ resulta

numa superfície contínua e fechada, predominantemente convexa, denominada superfície de interação.

Na fig. 4.3 é representada uma superfície de interação. A superfície interna pontilhada representa a parcela da resistência devida apenas ao concreto ($P_c - O$), e o espaço entre ela e a superfície de interação, representa a parcela devida apenas à armadura ($P_u - P_c$); a resistência da seção de concreto armado obtém-se pela soma destas duas parcelas ($P_u - O = (P_c - O) + (P_u - P_c)$).

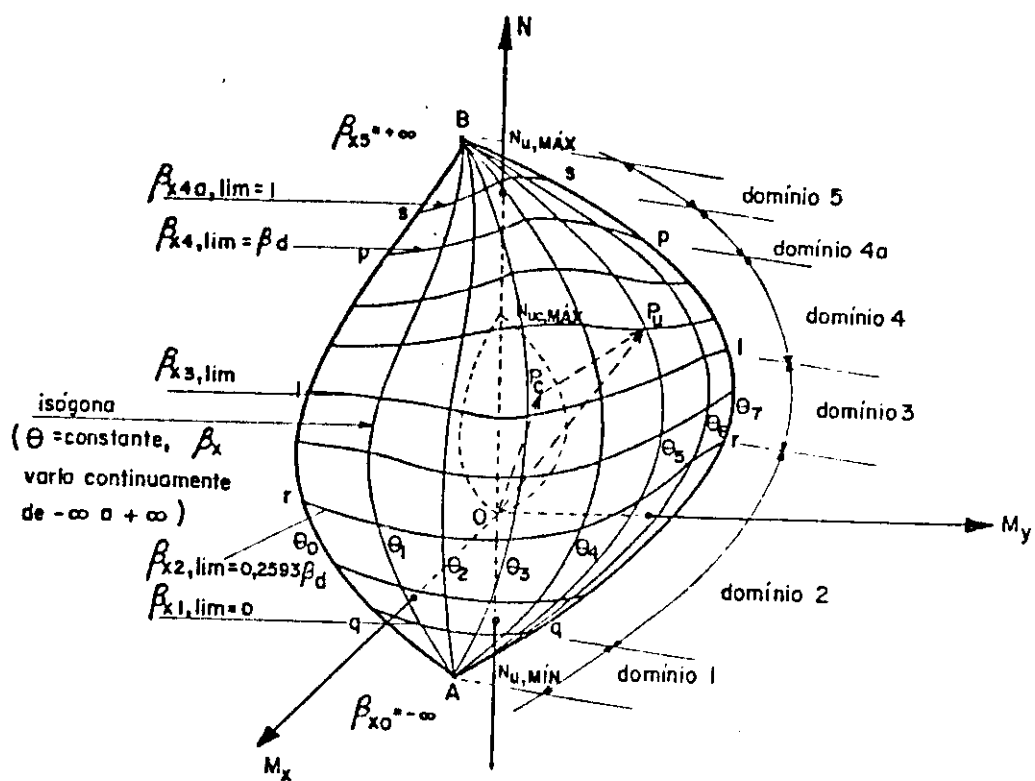


Fig. 4.3- Superfície de interação de uma seção qualquer de concreto armado: representa a resistência da seção a solicitações normais (FORNI, 1980)

Demonstra-se que para uma mesma seção de concreto armado, mantidas a distribuição da armadura e as características mecânicas dos materiais, o incremento da área total de armadura $A_{s,tot}$ se traduz numa expansão tridimensional da correspondente superfície de interação (Forni, 1980).

Desta propriedade advém que, conhecida a superfície de interação de uma seção com $\bar{A}_{s,tot}$ não nula, obtém-se todas as demais superfícies de interação para qualquer valor atribuído à área total de

armadura ($A_{s,tot}$), bastando para isso multiplicar cada um dos vetores $(\bar{P}_u - P_c)$ da superfície conhecida, pela relação das áreas das $A_{s,tot}/\bar{A}_{s,tot}$. Tem-se assim:

$$(P_d - P_c) = \frac{A_{s,tot}}{\bar{A}_{s,tot}} (\bar{P}_u - P_c) \quad (4.28)$$

4.5. DIMENSIONAMENTO ANALÍTICO DE SEÇÕES À FCO

O dimensionamento analítico consiste em fixar-se um valor $\bar{A}_{s,tot}$ para a área total de armadura e determinar-se $A_{s,tot,nec}$ através de um processo iterativo nas variáveis θ e $1/r$, cujo objetivo é fazer com que a semi-reta de origem P_c e que contém $(\bar{P}_u - P_c)$ (fig. 4.4) contenha o ponto P_d , de coordenadas (N_d, M_{xd}, M_{yd}) . Quando isto ocorrer ter-se-á:

$$(P_d - P_c) = \frac{A_{s,tot,nec}}{\bar{A}_{s,tot}} (\bar{P}_u - P_c) \quad (4.29)$$

$$A_{s,tot,nec} = \bar{A}_{s,tot} \frac{|(P_d - P_c)|}{|(\bar{P}_u - P_c)|} \quad (4.30)$$

$$A_{s,tot,nec} = \bar{A}_{s,tot} \frac{\sqrt{(N_d - N_{uc})^2 + (M_{xd} - M_{xuc})^2 + (M_{yd} - M_{yuc})^2}}{\sqrt{\bar{N}_{us}^2 + \bar{M}_{xus}^2 + \bar{M}_{yus}^2}} \quad (4.31)$$

Chamando-se de α_s o ângulo formado pelos vetores $(P_d - P_c)$ e $(\bar{P}_u - P_c)$, a semi-reta contém P_d quando $\alpha_s = 0$, isto é, quando:

$$f_s = 1 - \cos \alpha_s = 0 \quad (4.32)$$

Se P_d for interno ou estiver contido na superfície resistente do concreto, apenas o concreto da seção resiste à solicitação, isto é, $A_{s,tot,nec} = 0$. Assim, antes de se iniciar o processo iterativo para a determinação de $A_{s,tot,nec}$, deve-se determinar a posição relativa entre o ponto P_d e a superfície resistente do concreto.

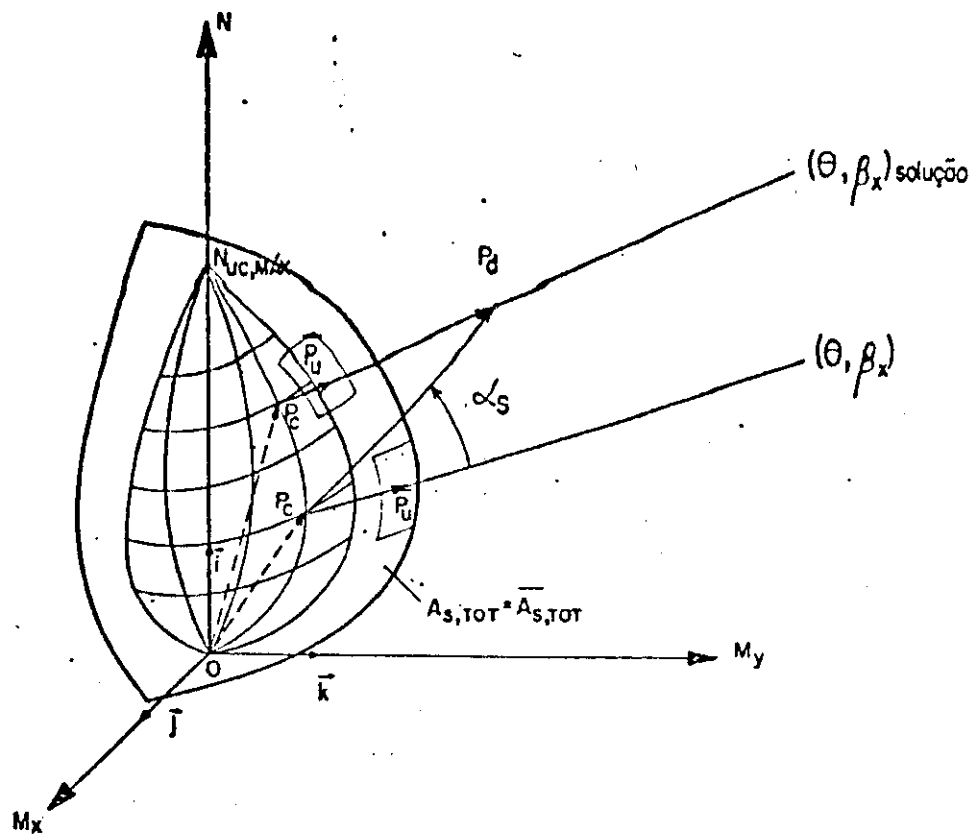


Fig. 4.4-Processo iterativo para o dimensionamento analítico de seções quaisquer sujeitas à flexão composta oblíqua (FORNI, 1980).

Se $N_d \leq 0$ ou $N_d > N_{uc,max}$, P_d é externo à superfície resistente do concreto, devendo-se proceder à determinação iterativa de $A_{s,tot\ nec}$,

Se $0 < N_d \leq N_{uc,max}$, P_d pode assumir qualquer posição relativamente à superfície resistente do concreto. Neste caso, a determinação da posição relativa pode ser feita por um processo iterativo nas variáveis θ e $1/r$, cujo objetivo é fazer com que a reta que contém o vetor $(P_c - 0)$ (Fig. 4.5) contenha P_d , pois quando isso acontecer, ter-se-á:

$$(P_d - 0) = t (P_c - 0) \quad (4.33)$$

, isto é:

$$N_d = t N_d \quad (4.34)$$

$$M_{xd} = t M_{xuc} \quad (4.35)$$

$$M_{yd} = t M_{yuc} \quad (4.36)$$

Onde o valor de t , calculado por exemplo de (4.34), define a posição relativa.

Assim:

- se $t < 1$, P_d é interno à superfície resistente do concreto e $A_{s,tot\ nec} = 0$.

- se $t = 1$, P_d está contido na superfície resistente do concreto e $A_{s,tot\ nec} = 0$.

- se $t > 1$, P_d é externo à superfície resistente do concreto, devendo-se proceder à determinação iterativa de $A_{s,tot\ nec}$.

Chamando de α_c o ângulo formado pelos vetores $(P_d - 0)$ e $(P_c - 0)$ (Fig. 4.5), a reta vetor $(P_c - 0)$ contém P_d quando $\alpha_c = 0$, isto é, quando:

$$f_c = 1 - \cos \alpha_c = 0 \quad (4.37)$$

Portanto, o objetivo deste processo iterativo é anular f_c , calculando-se então t , que define a posição relativa.

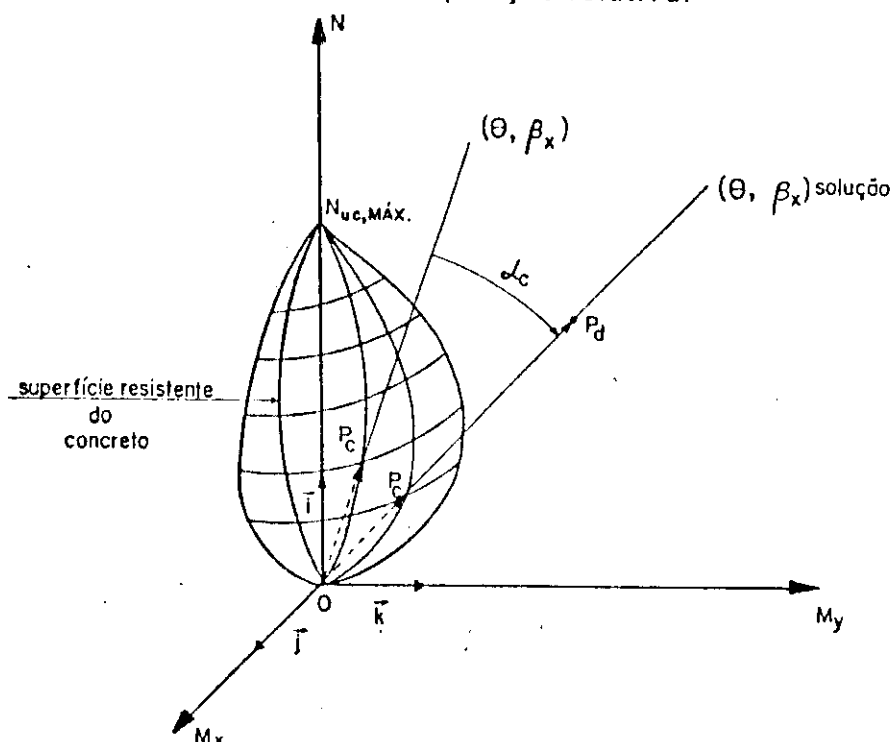


Fig. 4.5 - Processo iterativo para a determinação da posição relativa entre P_d e a superfície resistente do concreto quando $0 < N_d \leq N_{uc,máx}$ (FORNI, 1980).

Tendo-se em conta que :

$$\cos \alpha_s = \frac{(P_d - P_c) \cdot (\bar{P}_u - P_c)}{|(P_d - P_c)| \cdot |(\bar{P}_u - P_c)|} \quad (4.38)$$

$$\cos \alpha_c = \frac{(P_d - O) \cdot (P_c - O)}{|(P_d - O)| \cdot |(P_c - O)|} \quad (4.39)$$

E que:

$$(P_d - P_c) = (N_d - N_{uc}) \bar{i} + (M_{xd} - M_{xuc}) \bar{j} + (M_{yd} - M_{yuc}) \bar{k} \quad (4.40)$$

$$(\bar{P}_u - P_c) = \bar{N}_{us} \bar{i} + \bar{M}_{xus} \bar{j} + \bar{M}_{yus} \bar{k} \quad (4.41)$$

$$(P_d - O) = N_d \bar{i} + M_{xd} \bar{j} + M_{yd} \bar{k} \quad (4.42)$$

$$(P_c - O) = N_{uc} \bar{i} + M_{xuc} \bar{j} + M_{yuc} \bar{k} \quad (4.43)$$

Tem-se:

$$f_s = 1 - \frac{(N_d - N_{uc}) \bar{N}_{us} + (M_{xd} - M_{xuc}) \bar{M}_{xus} + (M_{yd} - M_{yuc}) \bar{M}_{yus}}{\sqrt{(N_d - N_{uc})^2 + (M_{xd} - M_{xuc})^2 + (M_{yd} - M_{yuc})^2} \cdot \sqrt{\bar{N}_{us}^2 + \bar{M}_{xus}^2 + \bar{M}_{yus}^2}} \quad (4.44)$$

$$f_c = 1 - \frac{N_d N_{uc} + M_{xd} M_{xuc} + M_{yd} M_{yuc}}{\sqrt{N_d^2 + M_{xd}^2 + M_{yd}^2} \sqrt{N_{uc}^2 + M_{xuc}^2 + M_{yuc}^2}} \quad (4.45)$$

Apresenta-se a seguir um processo iterativo de modificações sucessivas do par $(0, 1/r)$, adaptado do processo apresentado por Forni (1980), para o dimensionamento analítico de seções quaisquer de concreto armado com distribuição de armadura pré-fixada.

- 1) Verifica-se se $N_d \leq 0$. Em caso afirmativo, P_d é externo à superfície resistente do concreto, devendo-se ir para o passo 4; caso contrário, prosseguir.
- 2) Sendo $N_d > 0$, calcular $N_{uc, \max}$ correspondente à compressão uniforme da seção e compará-lo com N_d .

- se $N_d > N_{uc,m\acute{a}x}$, P_d é externo à superfície resistente do concreto, devendo-se ir para o passo 4;

- se $N_d = N_{uc,m\acute{a}x}$, deve-se verificar se M_{xd} e M_{yd} são ambos nulos. Em caso afirmativo, P_d está contido na superfície resistente do concreto, sendo $A_{s,tot,nec} = 0$, encerrando-se os cálculos; caso contrário, P_d é externo à superfície resistente do concreto, devendo-se ir para o passo 4;

- se $N_d < N_{uc,m\acute{a}x}$, prosseguir.

3) Sendo $0 < N_d < N_{uc,m\acute{a}x}$ a determinação da posição relativa entre o ponto P_d e a superfície resistente do concreto faz-se a partir da determinação do valor de t , calculado de 4.34. Nesta expressão, N_{uc} corresponde ao ponto P_c obtido da intersecção da reta OP_d com a superfície resistente do concreto (Fig. 4.5). Sua determinação é feita iterativamente, conforme segue:

3.1) Adotam-se um par $(\theta, 1/r) = (\theta_o, 1/r_o)$ e um par inicial de incrementos $(\Delta\theta, \Delta 1/r) = (\Delta\theta_o, \Delta 1/r_o)$.

3.2) Consideram-se os pares $(\theta_i, 1/r_i)$, com $i = 1, 2, \dots, 9$:

$$(\theta_1, 1/r_1) = (\theta - \Delta\theta, 1/r + \Delta 1/r); \quad (\theta_2, 1/r_2) = (\theta, 1/r + \Delta 1/r);$$

$$(\theta_3, 1/r_3) = (\theta + \Delta\theta, 1/r + \Delta 1/r); \quad (\theta_4, 1/r_4) = (\theta - \Delta\theta, 1/r);$$

$$(\theta_5, 1/r_5) = (\theta, 1/r); \quad (\theta_6, 1/r_6) = (\theta + \Delta\theta, 1/r);$$

$$(\theta_7, 1/r_7) = (\theta - \Delta\theta, 1/r - \Delta 1/r); \quad (\theta_8, 1/r_8) = (\theta, 1/r - \Delta 1/r);$$

$$(\theta_9, 1/r_9) = (\theta + \Delta\theta, 1/r - \Delta 1/r);$$

Para cada um dos pares, determinam-se os correspondentes ternos $(N_{uci}, M_{xuci}, M_{yuci})$ relativo

apenas à parcela resistida pelo concreto e calcula-se, através de 4.45 o correspondente valor de f_{ci} .

3.3) Determina-se o menor destes valores:

$$f_{cj} = \min (f_{ci}) \quad , \quad i = 1, 2, \dots, 9$$

3.4) Se $f_{cj} < f_o$ (valor muito pequeno, fixado), dá-se o processo por encerrado: $N_{uc} = N_{ucj}$ e $t = N_d/N_{uc}$.

Se $t \leq 1$, P_d é interno ou está sobre a superfície resistente do concreto, sendo $A_{s,tot,nec} = 0$, encerrando-se o procedimento.

Se $t > 1$, P_d é externo à superfície, devendo-se ir para o passo 4

3.5) Se $f_{cj} \geq f_o$, sendo $j \neq 5$, faz-se $(\theta, 1/r) = (\theta_j, 1/r_j)$ e volta-se ao passo 3.2.

se $j=5$ \Rightarrow faz-se $(\Delta\theta, \Delta 1/r) = (p_1 \Delta\theta, p_1 \Delta 1/r)$

se $j=2$ ou $j=8$ \Rightarrow faz-se $(\Delta\theta, \Delta 1/r) = (\Delta\theta, p_1 \Delta 1/r)$

se $j=4$ ou $j=6$ \Rightarrow faz-se $(\Delta\theta, \Delta 1/r) = (p_1 \Delta\theta, \Delta 1/r)$

sendo sempre $0 < p_1 < 1$ e volta-se ao passo 3.2.

4) Com P_d externo à superfície resistente do concreto, determina-se $A_{s,tot,nec}$ como segue:

4.1) Adota-se ρ , valor da taxa máxima de armadura para a seção dada e calcula-se o valor de $\bar{A}_{s,tot}$.

4.2) Determinam-se os valores máximo ($N_{u,máx}$) e mínimo ($N_{u,mín}$) da força normal capaz de ser resistida pela

seção. Se a normal atuante (N_d) não estiver no intervalo definido por ($N_{u,min}$) e ($N_{u,max}$), interrompe-se o procedimento.

4.3) Adotam-se um par $(\theta, 1/r) = (\theta_o, 1/r_o)$ e um par inicial de incrementos $(\Delta\theta, \Delta 1/r) = (\Delta\theta_o, \Delta 1/r_o)$.

4.4) Consideram-se os pares $(\theta_i, 1/r_i)$, com $i = 1, 2, \dots, 9$:

$$\begin{aligned} (\theta_1, 1/r_1) &= (\theta - \Delta\theta, 1/r + \Delta 1/r); & (\theta_2, 1/r_2) &= (\theta, 1/r + \Delta 1/r); \\ (\theta_3, 1/r_3) &= (\theta + \Delta\theta, 1/r + \Delta 1/r); & (\theta_4, 1/r_4) &= (\theta - \Delta\theta, 1/r); \\ (\theta_5, 1/r_5) &= (\theta, 1/r); & (\theta_6, 1/r_6) &= (\theta + \Delta\theta, 1/r); \\ (\theta_7, 1/r_7) &= (\theta - \Delta\theta, 1/r - \Delta 1/r); & (\theta_8, 1/r_8) &= (\theta, 1/r - \Delta 1/r); \\ (\theta_9, 1/r_9) &= (\theta + \Delta\theta, 1/r - \Delta 1/r); \end{aligned}$$

Para cada um dos pares, determinam-se os correspondentes ternos $(N_{uc,i}, M_{xu,i}, M_{yu,i})$ e $(N_{us,i}, M_{xus,i}, M_{yus,i})$. Através de 4.44, calcula-se o correspondente valor de f_{sj} .

4.5) Determina-se o menor destes valores:

$$f_{sj} = \min (f_{si}), \quad i = 1, 2, \dots, 9$$

4.6) Se $f_{sj} < f_o$ (valor muito pequeno, fixado), dá-se o processo por encerrado e calcula-se $A_{s,tot,nec}$ pela expressão 4.31. O procedimento está encerrado

4.7) Para $f_{sj} \geq f_o$,

$$\text{se } j=5 \quad \Rightarrow \text{faz-se } (\Delta\theta, \Delta 1/r) = (p_1 \Delta\theta, p_1 \Delta 1/r)$$

$$\text{se } j=2 \text{ ou } j=8 \quad \Rightarrow \text{faz-se } (\Delta\theta, \Delta 1/r) = (\Delta\theta, p_1 \Delta 1/r)$$

$$\text{se } j=4 \text{ ou } j=6 \quad \Rightarrow \text{faz-se } (\Delta\theta, \Delta 1/r) = (p_1 \Delta\theta, \Delta 1/r)$$

Nos demais casos não há alteração.

Adota-se p_1 no intervalo $0 < p_1 < 1$. É satisfatório adotar-se $p_1 = 0,5$.

8) Faz-se $(\theta, 1/r) = (\theta_j, 1/r_j)$

9) Retorna-se ao passo 4

4.6. DETERMINAÇÃO DO ESTADO DE DEFORMAÇÕES E DE TENSÕES

Neste caso, trata-se de resolver o primeiro problema básico. A seção está totalmente definida e o estado de deformações ou de tensões fica determinado pelo terno $(\theta, \beta_x, 1/r)$ o qual deve ser obtido iterativamente.

Considere a fig. 4.6. Nela esta representada a relação $Nd \times 1/r$. Os pontos pertencentes à curva, representam para uma dada inclinação θ da linha neutra, os ternos $(Nd, Md, 1/r)$ que conduzem a seção a um estado limite último, enquanto que os pontos internos a curva representam os ternos $(Nd, Md, 1/r)$ para os quais a seção não atinge um estado limite último.

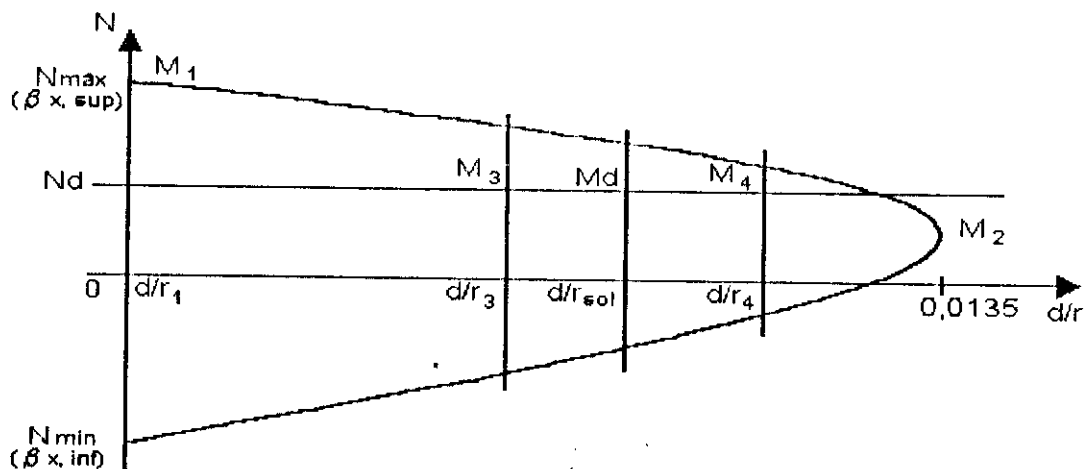


Fig. 4.6 - Diagrama $N \times 1/r$ para uma dada inclinação θ da linha neutra de uma seção qualquer

Adotando-se um valor constante para a inclinação θ da linha neutra de uma seção qualquer, o estado de deformações desta seção fica determinado conhecendo-se os valores de sua curvatura e da profundidade adimensional de sua linha neutra, $1/r$ e β_x respectivamente.

Desta forma, através de um processo iterativo, apresentado na fig. 4.6, pode-se determinar o estado de deformações de uma seção qualquer. Este processo consiste em adotar-se sucessivamente valores para a curvatura da seção em estudo e determinar-se o valor de β_x de tal forma que a força normal resistente equilibre a normal de cálculo N_d . Nesta situação verifica-se o valor do momento fletor resistente e compara-se com o momento fletor atuante. Se estes forem iguais a menos de uma tolerância, o estado de tensões determinado é o procurado. No entanto, se os momentos fletores não coincidirem, deve-se prosseguir com as iterações adotando-se um novo valor para $1/r$, interpolando-se linearmente os dois últimos valores obtidos para $1/r$ e para o momento fletor resistente.

Na fig. 4.6., adotou-se inicialmente a curvatura mínima $1/r_1=0$ e máxima $1/r_2=0.0135 d$. Como a curvatura máxima não permitiu o equilíbrio de N_d , adotou-se a curvatura $1/r_3 = 1/r_2 / 2$. Com este novo valor da curvatura foi possível equilibrar N_d e obter-se um momento fletor resistente M_{x3} , o qual é menor que M_d , como se observa pela figura. Interpolando-se linearmente os valores de $1/r_2$ e $1/r_3$ e seus respectivos momentos fletores, obtém-se $1/r_4$ e M_4 . Repetindo-se sucessivamente este procedimento, obtém-se $1/r_{sol}$ e $\beta_{x,sol}$ que caracterizam o estado de deformações da seção para os esforços N_d e M_d .

4.6.1. PROCESSO ITERATIVO PARA EQUILIBRAR AS TENSÕES NORMAIS ATUANTE E RESISTENTE.

Inicialmente será apresentado um algoritmo que para um par $(\theta, 1/r)$, determina o valor da profundidade adimensional da linha neutra (β_x) que conduz ao equilíbrio entre as forças normais atuante e resistente (Paula, 1988), o qual é parte integrante do algoritmo utilizado na determinação do estado de deformações da seção.

1. Adota-se um par $(\theta, 1/r)$ para o qual será obtido o valor de β_x que conduz ao equilíbrio entre as forças normais atuante e resistente.

2. Definem-se os valores extremos do intervalo, para variação da profundidade da linha neutra.

2.1. Limite inferior de β_x : Considera-se para limite inferior de β_x a expressão (3.9):

$$\beta_{xl} = 1 - 0,01 / (d(\theta)/r)$$

2.2. Limite superior de β_x :

Considera-se para limite superior de β_x , caso este seja menor ou igual a $h(\theta)/d(\theta)$, a expressão 3.10:

$$\beta_{xs} = 0,0035 / (d(\theta)/r)$$

Ou, caso contrário, a expressão 3.11:

$$\beta_{xs} = 0,002/(d(\theta)/r) + (3/7)/(h(\theta)/d(\theta))$$

3. Calculam-se as forças normais resistentes, N_1 e N_2 , correspondentes aos limites, inferior e superior de β_x , respectivamente, usando-se a expressão 4.1.

4. Verifica-se a normal N_{fix} , pertencente ao intervalo definido no passo 3.

4.1. Se $N_{fix} < N_1$ ou $N_{fix} > N_2$, o equilíbrio é impossível, uma vez que os limites de resistência são ultrapassados. Neste caso, adotam-se novos valores para θ e $1/r$.

4.2. Se N_{fix} for igual ou aproximadamente igual, a menos de uma tolerância, a N_1 ou N_2 , prossegue-se no passo 9. Caso contrário, vai-se ao passo 5.

5. Calcula-se um novo valor para β_x , interpolando-se linearmente os valores β_{xi} e β_{xs} .

$$\beta_x = \frac{(N_{fix} - N_1)(\beta_{xs} - \beta_{xi})}{N_2 - N_1} + \beta_{xi} \quad (4.46)$$

6. Com o valor de β_x calculado em 5, obtem-se um novo valor para a força normal interna, N_i .

7. Verificar o equilíbrio

7.1. Se $N_i = N_{fix}$, a menos de uma tolerância pré-fixada, vai-se para o passo 9.

7.2. Se o número de iterações extrapolar um valor fixado, deve-se adotar nova curvatura.

8. Substituindo-se convenientemente N_1 ou N_2 pelo valor de N_i calculado em 6, e β_{xi} ou β_{xs} pelo valor de β_x , calculado em 5, de forma que a força normal fixada se encontre dentro do novo intervalo, retorna-se ao passo 5.

9. Calcula-se o momento fletor resistente.

10. Retorna-se ao passo 1 para novo valor de $1/r$.

4.6.2. PROCESSO ITERATIVO PARA DETERMINAÇÃO DO ESTADO DE DEFORMAÇÕES DA SEÇÃO

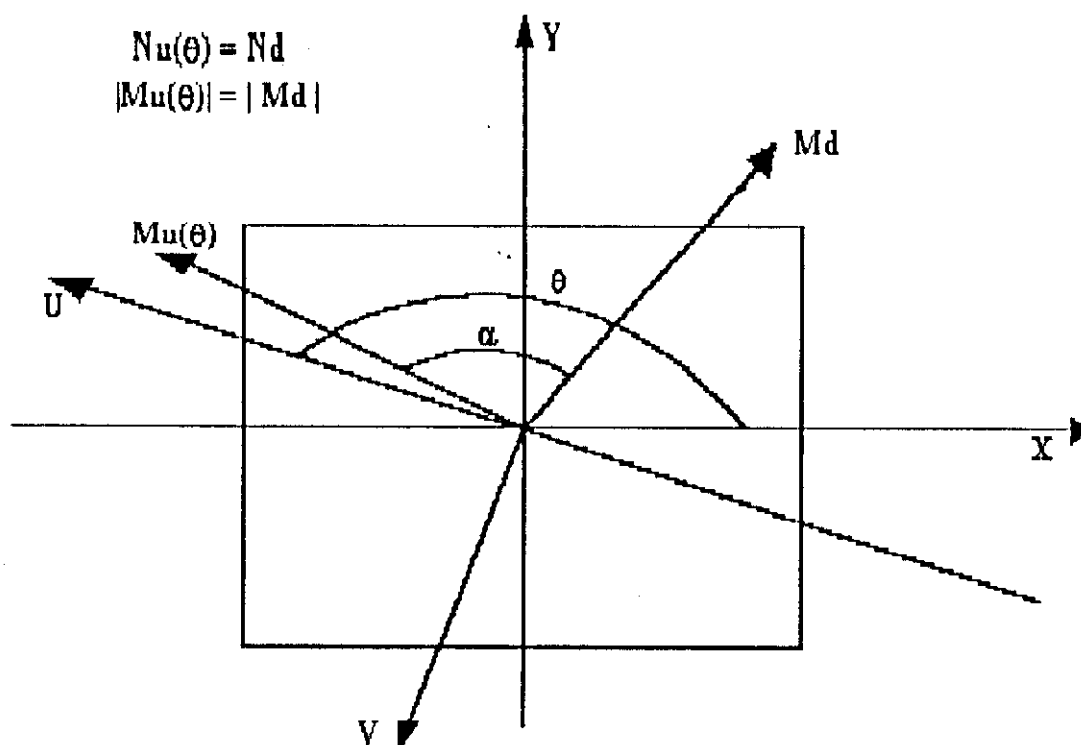
Na flexão oblíqua composta, o estado de deformações correspondente ao terno de solicitações de cálculo (N_d , M_{xd} , M_{yd}), é obtido quando o vetor de esforços resistentes da seção de componentes (N_u , M_{xu} , M_{yu}) for equivalente ao vetor de esforços solicitantes, ou seja, quando:

$$N_u = N_d \quad (4.47)$$

$$M_{xu} = M_{xd} \quad (4.48)$$

$$M_{yu} = M_{yd} \quad (4.49)$$

O processo iterativo que conduz à solução deste problema, consiste na determinação para uma dada inclinação θ da linha neutra, de um estado de deformações que satisfaça a equação 4.47, a partir do apresentado no ítem 4.6.1, e, a seguir, verificar se nesta situação são satisfeitas as equações 4.48 e 4.49 (Fig. 4.7).



ig. 4.7 - Processo iterativo para determinar o estado de deformações de uma seção qualquer.

Esta verificação é feita indiretamente, considerando-se:

$$M_u = (M_{xu}, M_{yu}) \quad (4.50)$$

$$M_d = (M_{xd}, M_{yd}) \quad (4.51)$$

E, portanto:

$$|M_u| = (M_{xu}^2 + M_{yu}^2)^{0.5} \quad (4.52)$$

$$|M_d| = (M_{xd}^2 + M_{yd}^2)^{0.5} \quad (4.53)$$

Sendo α o ângulo formado entre os vetores M_u e M_d , ficam verificadas as equações 4.48 e 4.49 quando ocorrer simultaneamente:

$$|M_u| = |M_d|$$

$$\alpha = 0$$

Para que α seja nulo deve-se ter $\cos \alpha = 1$ e, portanto, $f_\alpha = 1 - \cos \alpha$ também nulo. Analogamente ao apresentado no item 4.5, tem-se então que:

$$f_\alpha = 1 - \frac{M_{ux} \cdot M_{dx} + M_{uy} \cdot M_{dy}}{\sqrt{M_{ux}^2 + M_{uy}^2} \cdot \sqrt{M_{dx}^2 + M_{dy}^2}} \quad (4.54)$$

deve ser nulo.

Portanto o processo iterativo visa determinar dentre todos os possíveis vetores M_u com norma igual a do vetor M_d , aquele que anula a função f_α . Faz-se necessário a adoção de um método para variar o valor da inclinação θ da linha neutra, de tal forma que f_α convirja para um valor nulo. Colocando-se f_α em função de θ tem-se:

$$f_\alpha = f(\theta) \quad (4.55)$$

Adotando-se como uma aproximação de $f(\theta)$ um polinômio do segundo grau (Fig. 4.8), obtém-se:

$$f(\theta) = A \theta^2 + B \theta + C \quad (4.56)$$

Para que os parâmetros A, B e C fiquem determinados é necessário que sejam conhecidos inicialmente os valores de $f(\theta)$ e conseqüentemente de f_α para três valores quaisquer de θ . Assim:

$$DT = \theta_1^2 \cdot (\theta_2 - \theta_3) + \theta_2^2 \cdot (\theta_3 - \theta_1) + \theta_3^2 \cdot (\theta_1 - \theta_2) \quad (4.57)$$

$$A = (f_{\alpha 1} \cdot (\theta_2 - \theta_3) + f_{\alpha 2} \cdot (\theta_3 - \theta_1) + f_{\alpha 3} \cdot (\theta_1 - \theta_2)) / DT \quad (4.58)$$

$$B = (f_{\alpha 1} \cdot (\theta_3^2 - \theta_2^2) + f_{\alpha 2} \cdot (\theta_1^2 - \theta_3^2) + f_{\alpha 3} \cdot (\theta_2^2 - \theta_1^2)) / DT \quad (4.59)$$

$$C = (f_{\alpha 1} \cdot (\theta_2^2 \cdot \theta_3 - \theta_3^2 \cdot \theta_2) + f_{\alpha 2} \cdot (\theta_3^2 \cdot \theta_1 - \theta_1^2 \cdot \theta_3) + f_{\alpha 3} \cdot (\theta_1^2 \cdot \theta_2 - \theta_2^2 \cdot \theta_1)) / DT \quad (4.60)$$

A equação 4.56 terá um valor mínimo quando:

$$\theta_i = - B / 2A \quad (4.61)$$

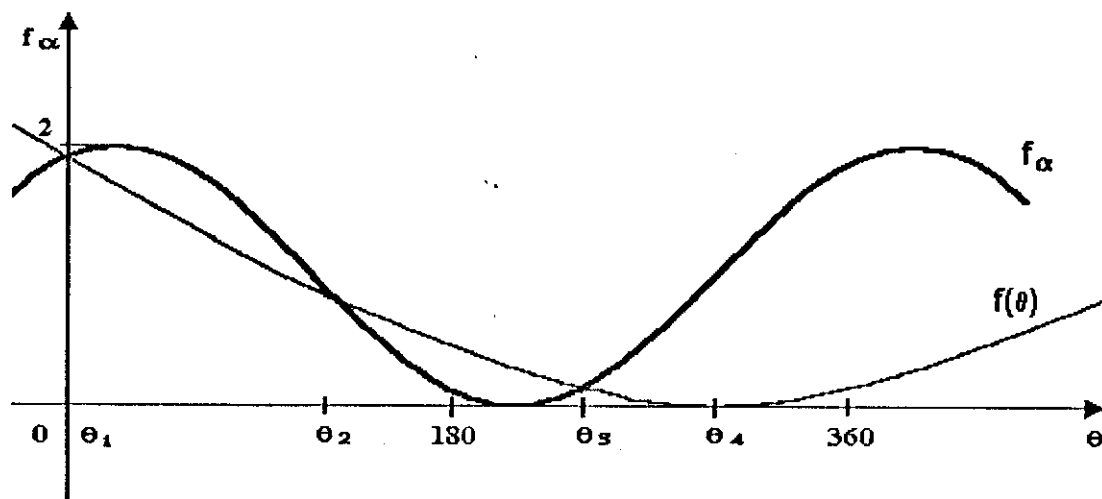


Fig. 4.8 - Funções f_{α} e $f(\theta)$

Caso não seja possível anular $f_{\alpha i}$ a menos de uma tolerância para este valor de θ_i , recalculam-se os parâmetros A, B e C, utilizando-se o valor obtido para $f_{\alpha i}$, repetindo-se o procedimento, até que este se anule.

Apresenta-se a seguir um algoritmo que utiliza este procedimento para determinar o estado de deformações de uma seção qualquer sujeita a flexão composta oblíqua. Este algoritmo foi desenvolvido a partir do utilizado por Paula (1988) na solução de problemas da flexão normal composta.

1. Adotam-se três valores iniciais quaisquer (θ_1 , θ_2 , θ_3) para a inclinação da linha neutra em relação ao eixo X.

Sugere-se adotar $\theta_1 = 0^\circ$, $\theta_2 = 120^\circ$, $\theta_3 = 240^\circ$. Para cada um dos três valores iniciais de θ , executa-se o procedimento descrito a seguir:

2. Com o valor assumido para θ , procura-se determinar a curvatura capaz de produzir o equilíbrio dos esforços atuantes.

2.1. Adota-se a curvatura mínima adimensional :

$$(1/r)_1 = 0 \quad (4.62)$$

Usando-se o algoritmo apresentado em 4.6.1 obtém-se, caso o procedimento não seja interrompido nos passos 4.1 ou 7.2 do algoritmo citado, além de um esforço normal resistente N_1 , que equilibra o esforço normal atuante, o momento fletor resistente M_1 da seção em análise.

2.2. Se o momento fletor resistente M_1 , calculado em 2.1, equilibrar o momento fletor atuante, vai-se para o passo 2.10.

2.3. Adota-se em seguida a curvatura máxima adimensional, para a seção em análise, admitindo-se que, a deformação última de tração no aço, 10 ‰, e a deformação última de compressão no concreto, 3,5 ‰ sejam atingidas ao mesmo tempo, isto é:

$$(1/r)_2 = 0.0135 / d \quad (4.63)$$

Considerando-se o valor de $(1/r)_2$, e usando-se o algoritmo descrito no item 4.6.1, obtém-se, caso o procedimento não seja interrompido no passo 4.1 ou 7.2 do citado algoritmo, além de um esforço normal resistente N_2 , que equilibra o esforço normal atuante, o momento fletor resistente M_2 , passando-se ao passo

2.5 deste algoritmo. Caso não seja possível a determinação da força normal, como indicado no passo 4.1, do algoritmo acima citado, passar ao passo 2.4 deste.

2.4. A partir dos valores $(1/r)_1$ e $(1/r)_2$ obtém-se um novo valor para $(1/r)_2$, usando-se a expressão:

$$(1/r)_2 = (1/r)_1 / 2 + (1/r)_2 / 2 \quad (4.64)$$

A verificação do novo valor de $(1/r)_2$ é feita como no passo 2.3. Novos valores de $(1/r)_2$, calculados pela média dada acima, devem se repetir até que o equilíbrio entre esforço normal atuante e resistente seja encontrado. Determina-se, em seguida, segundo o passo acima citado, o momento fletor M_2 .

2.5. Se o momento fletor resistente M_2 equilibrar o momento fletor atuante na seção em análise, vai-se para o passo 2.10.

2.6. Interpolando-se linearmente os valores $(1/r)_1$, M_1 , $(1/r)_2$ e M_2 , juntamente com o momento fletor M , atuante na seção, chega-se à expressão:

$$(1/r)_2 = (1/r)_2 - [(1/r)_2 - (1/r)_1] \cdot (M_2 - M) / (M_2 - M_1) \quad (4.65)$$

a qual fornece um novo valor para $(1/r)_2$. Caso o número de iterações ultrapasse um valor estabelecido, faz-se :

$$\theta_i = (\theta_i + \theta_{i+1}) / 2 \quad (4.66)$$

2.7. Se $(1/r)_2 \cong (1/r)_1$, faz-se:

$$\theta_i = (\theta_i + \theta_{i+1}) / 2 \quad (4.66)$$

e retorna-se ao passo 2.

- 2.8. Retorna-se ao passo 2.3, mantendo-se o valor de $(1/r)_2$
- 2.9. Usando-se a expressão 4.42, determina-se o valor de f_α . Se este for nulo, a menos de uma tolerância estabelecida, o estado de deformações da seção está determinado e encerra-se o procedimento.
- 2.10. Caso não se tenha determinado pelo menos três valores para f_α , adota-se o próximo valor pré-definido para o ângulo θ e retorna-se ao passo 2.
- 2.11. Utilizando-se três valores de θ e os valores respectivos determinados para f_α , calculam-se os parâmetros A, B e C de acordo com as equações 4.58, 4.59 e 4.60. Caso A, B e C estejam sendo calculados pela primeira vez, θ_1 , θ_2 e θ_3 , manterão os valores definidos no passo 1. Senão, θ_2 assumirá o valor determinado para θ no passo 2.12, θ_1 e θ_3 assumirão respectivamente os valores imediatamente inferior e superior a θ , utilizados na iteração anterior.
- 2.12. Calcula-se um novo valor para θ utilizando-se a equação 4.61 e retorna-se ao passo 2.

4.7. DEFORMAÇÕES NA FLEXÃO COMPOSTA OBLÍQUA

Em uma barra reta submetida à FCO, seu eixo sofre deformações. Transversalmente, os deslocamentos dão origem as excentricidades e_2 de segunda ordem (figura 4.5), as quais devem ser levadas em conta no estudo do equilíbrio da peça, principalmente quando esta for esbelta.

Ao longo da barra, o plano de flexão, que não é perpendicular ao plano que contém o momento fletor, varia de seção

para seção, tornando o eixo da barra uma curva reversa. Processos de verificação da estabilidade de barras sujeitas à FCO que desconsideram este fato, como no caso do Processo do Pilar Padrão, podem apresentar resultados contra a segurança (Marcolti,1984).

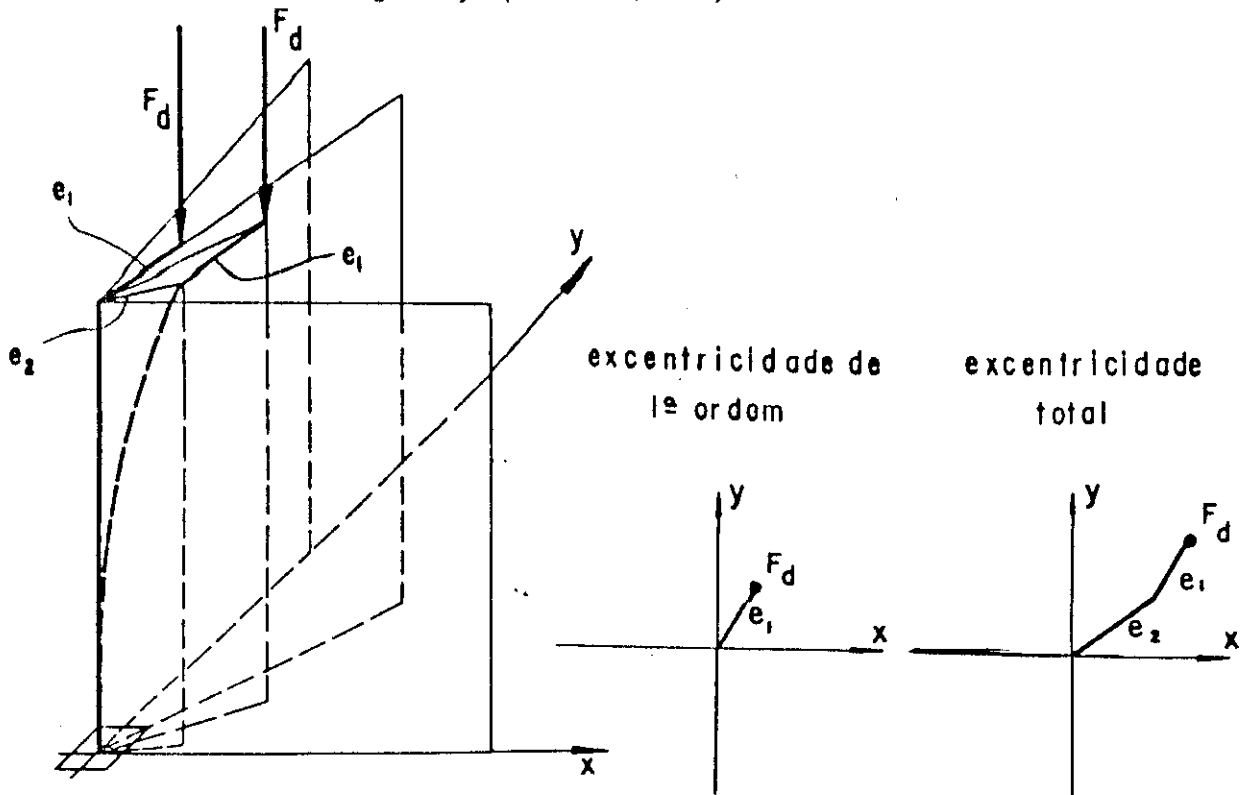


fig. 4.5 - Deformações em uma barra reta sujeita à flexão composta oblíqua (FUSCO,1981)

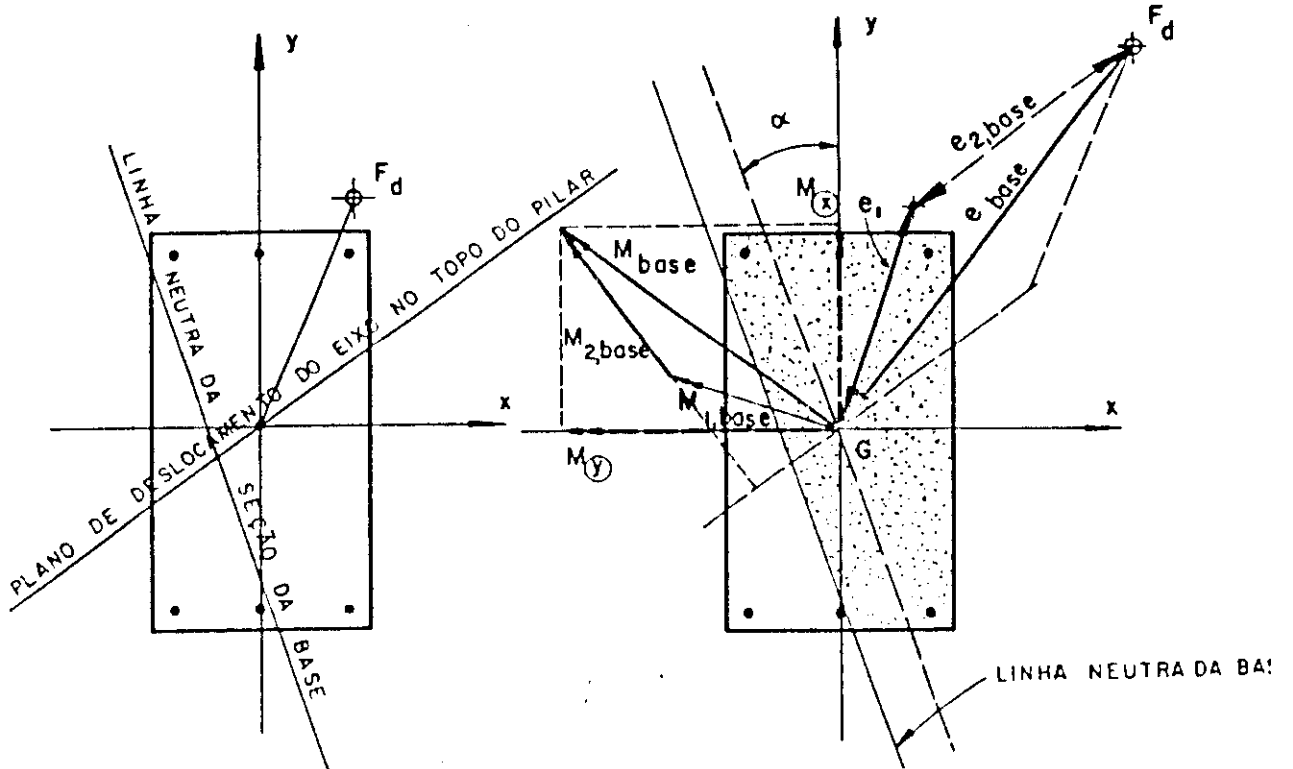


fig. 4.6 - Excentricidades de primeira e segunda ordem e os esforços solicitantes na seção da base da barra (FUSCO,1981)

A excentricidade e_1 de primeira ordem e e_2 de segunda ordem na seção da base da barra com seus respectivos momentos fletores são apresentados na fig. 4.6.

Para uma seção genérica i da barra tem-se:

$$N_d = F_d \quad (4.67)$$

$$| M_{1,i} | = F_d \cdot e_1 \quad (4.68)$$

$$| M_{2,i} | = F_d \cdot e_{2,i} \quad (4.69)$$

$$\vec{M}_{d,i} = \vec{M}_1 + \vec{M}_{2,i} \quad (4.70)$$

Portanto, o conhecimento das excentricidades com que a força normal de cálculo atua na seção e a sua intensidade, é suficiente para a determinação do momento fletor de cálculo atuante na seção. A excentricidade e_2 pode ser determinada utilizando-se o procedimento apresentado a seguir:

1. Determinam-se os esforços de primeira ordem atuantes em todas as seções da barra.
2. Determinam-se as curvaturas em todas as seções da barra, através da aplicação em cada seção do procedimento apresentado no item 4.6.2.
3. A seguir, pode-se utilizar a Regra dos Trapézios para proceder-se a integração das curvaturas ao longo da barra e assim obter-se as excentricidades de segunda ordem para uma determinada iteração (Paula, 1988) .
4. Se alguma das excentricidades determinadas tiver sofrido um incremento maior do que uma tolerância adotada, recalculam-se os esforços totais atuantes em todas as

seções e retorna-se ao passo 2. Caso contrário encerra-se o procedimento.

As excentricidades de segunda ordem podem ser estimadas exageradamente, se na determinação do estado de deformações, forem adotadas para os materiais, as mesmas resistências de cálculo utilizadas para o dimensionamento no Estado Limite Último. Portanto, neste caso, pode-se adotar para os coeficientes de minoração das resistências dos materiais os seguintes valores: $\gamma_c = 1,2$ e $\gamma_s = 1,05$ (Marcotti, 1984).

CAPÍTULO 5

ESTUDO DE PILARES SUBMETIDOS À F.C.O.

5.1. DIMENSIONAMENTO DE PILARES COM A CONSIDERAÇÃO DA NÃO-LINEARIDADE FÍSICA E GEOMÉTRICA

Considere-se um pilar em concreto armado, isolado, sujeito a carregamentos permanentes e acidentais, composto por barras de seção qualquer, nas quais são conhecidas a distribuição das armaduras, mas não a área destas. O dimensionamento deste pilar utilizando o Método Geral apresentado por Fusco (1980), deve levar em conta tanto a não linearidade física do material como a não linearidade geométrica decorrente das deformações de segunda ordem.

A não linearidade física, como consequência das relações não lineares entre σ e ϵ do concreto e do aço, pode ser levada em conta, conhecendo-se as relações momento-interno x força normal x curvatura para todas seções nas quais o pilar foi discretizado. Estas relações podem ser obtidas através dos procedimentos apresentados no Capítulo 4.

A não linearidade geométrica é considerada levando-se em conta no dimensionamento do pilar além dos esforços iniciais devidos às cargas aplicadas a ele e os esforços devidos as excentricidades acidentais, também os momentos fletores de segunda ordem, que aparecem em virtude das deformações do pilar.

Estes esforços de segunda ordem só podem ser determinados iterativamente, utilizando-se em uma iteração os deslocamentos obtidos na iteração anterior. A cada iteração ficam determinados os deslocamentos do pilar se, tendo este sido discretizado em um número qualquer de pontos, forem conhecidas as curvaturas

correspondentes às seções transversais de cada um destes pontos, o que pode ser feito utilizando-se o procedimento apresentado no item 4.6.

Deve-se repetir esta análise até que as deformações convirjam para um valor máximo limite, o qual, se atingido, comprova a segurança do pilar; ou até que este ultrapasse um estado limite último.

No entanto, deseja-se dimensionar o pilar e, portanto, não se conhece a priori a seção da armadura necessária para que esta se encontre na primeira situação. Desta forma, deve-se fazer uso de um segundo processo iterativo o qual consiste em determinar-se sucessivamente a armadura mínima necessária em cada seção do pilar, utilizando-se o processo apresentado no item 4.5, para que estas não ultrapassem o estado limite último quando o pilar sofrer os máximos deslocamentos de segunda ordem.

Este segundo processo iterativo termina quando a seção de armadura determinada em uma iteração for igual a menos de uma tolerância ao valor obtido na iteração anterior.

Apresenta-se a seguir um algoritmo que sintetiza os processos iterativos descritos aqui.

1) Determina-se para cada ponto em que o pilar foi discretizado, os esforços de primeira ordem atuantes neles.

2) Utiliza-se o procedimento apresentado no item 4.5 para determinar-se a armadura mínima necessária em cada seção do pilar, para que em nenhuma delas se ultrapasse o estado limite último. Se esta armadura for igual a menos de uma tolerância à armadura obtida na iteração anterior, o pilar está dimensionado.

3) Utilizando-se o procedimento apresentado no item 4.6.2. determinam-se as curvaturas em cada uma das seções. A partir destes valores, obtem-se os deslocamentos em todos os pontos do pilar. Caso alguma seção tenha ultrapassado o estado limite último, retorna-se ao

passo 2 utilizando-se desta vez os esforços determinados nesta iteração.

4) Recalculam-se os esforços solicitantes em todas as seções do pilar, levando-se em conta os deslocamentos obtidos no passo 3. Se os deslocamentos obtidos em uma determinada iteração forem iguais, a menos de uma tolerância estabelecida, aos obtidos na iteração anterior, o pilar é estável para o carregamento dado e para a armadura adotada. Retorna-se assim ao passo 2 utilizando-se os esforços obtidos nesta etapa. Caso contrário, vai-se ao passo 3.

5.2. SOLICITAÇÕES ATUANTES EM ESTRUTURAS DE EDIFÍCIOS

Os carregamentos básicos de edifícios são divididos em verticais e horizontais. Os verticais são separados em permanentes e de uso (NBR 6120), enquanto os horizontais são provenientes da ação do vento (NBR 6123).

Também quando se trabalha com estruturas de comportamento não linear, deve-se combinar as ações características afetadas por coeficientes de ponderação, estabelecendo-se tantas combinações quantas forem necessárias para que a segurança da estrutura seja verificada (NBR 8681). Para cada uma destas combinações, aplica-se o processo de dimensionamento apresentado em 5.1, adotando-se para cada barra a maior taxa de armadura determinada.

5.3. EXCENTRICIDADE ACIDENTAL

Para se levar em conta a incerteza da localização da força normal e o possível desvio do eixo da peça durante sua construção, em relação à posição prevista no projeto, a NBR 6118 prevê a consideração de uma excentricidade adicional de primeira ordem, designada por

excentricidade acidental e_a . O valor desta excentricidade é fornecida pela expressão 5.4, devendo ser considerada na direção em que ela puder ser mais desfavorável para a seção em estudo, sendo:

Para a flexão composta oblíqua:

$$e_a = h/30 \quad , \quad \text{onde } e_a \geq 2,0 \text{ cm} \quad (5.1)$$

Para compressão centrada:

$$e_{ax} = h_x/30 \quad , \quad \text{onde } e_{ax} \geq 2,0 \text{ cm} \quad (5.2)$$

$$e_{ay} = h_y/30 \quad , \quad \text{onde } e_{ay} \geq 2,0 \text{ cm} \quad (5.3)$$

devendo e_{ax} e e_{ay} serem tomados separadamente e adotando-se a situação mais desfavorável.

5.4. CONSIDERAÇÃO DA FLUÊNCIA

Na presença de cargas de longa duração, o concreto sofre aumento de deformações ao longo do tempo. Com isso são aumentados os momentos fletores de segunda ordem. Como a fluência ocorre sob a ação dos esforços permanentes característicos, as tensões no concreto são suficientemente baixas e, devido a isto, pode-se empregar a teoria linear da fluência, onde:

$$\varepsilon_{cc} = \phi \varepsilon_c \quad (5.4)$$

$$\varepsilon_{c,\text{total}} = \varepsilon_c + \varepsilon_{cc} = (1 + \phi) \varepsilon_c \quad (5.5)$$

Sendo:

ε_c = deformação imediata do concreto

ε_{cc} = deformação por fluência do concreto

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

ϕ = função de fluência

Embora seja adotada a teoria linear da fluência, não existe a linearidade física do sistema, não sendo válido, portanto, o princípio

da superposição de efeitos. Sendo assim, a fluência afeta apenas uma parcela das deformações que definem a curvatura, sendo então:

$$(1/r)_{total} = (\epsilon_s + \epsilon_{c,total}) / d = (\epsilon_s + (1 + \phi) \epsilon_c) / d \quad (5.6)$$

Devido à complexidade da determinação da função ϕ , Fusco (1980) apresenta um método aproximado para sua consideração, o qual pode ser aplicado mesmo no caso de pilares muito esbeltos ou de seção variável. De acordo com este método, realiza-se o cálculo como se toda a carga fosse de longa duração, adotando-se para a função de fluência o valor equivalente efetivo dado por:

$$\phi_{ef} = \alpha \cdot \beta \cdot \phi \quad (5.7)$$

onde:

α = fração da força normal que produz fluência

β = fração do momento fletor de primeira ordem que produz fluência

De acordo com Fusco (1980) pode-se adotar $\phi = 2$.

CAPÍTULO 6

PROGRAMA PARA O DIMENSIONAMENTO AUTOMÁTICO PILARES SUBMETIDOS A F.C.O.

6.1. INTRODUÇÃO

Com a finalidade de se automatizar os procedimentos de cálculo apresentados nos capítulos 4 e 5, foi elaborado um programa para computador denominado "DP" (Dimensionamento de Pilares). Este programa é constituído por três módulos: entrada e armazenamento de dados, dimensionamento de pilares e apresentação gráfica. Cada um deles foi desenvolvido na linguagem de programação que apresentou maior eficiência na realização das tarefas propostas. Assim o módulo de entrada e armazenamento de dados foi desenvolvido em Clipper 5.0, o módulo de dimensionamento em FORTRAN 77 e o módulo de apresentação gráfica em Turbo Basic.

Deve-se ressaltar que o programa "DP", desenvolvido para enfrentar problemas relacionados a flexão composta oblíqua em pilares com esbeltez elevada e de seção qualquer, mostrou-se também um instrumento de fácil utilização e eficiência na definição automatizada destes pilares e de suas seções transversais.

6.2. UTILIZAÇÃO DO PROGRAMA

Todas as funções possíveis de serem realizadas pelo programa podem ser acessadas através da linha de menus que é

permanentemente exibida na tela. Ela é composta pelos seguintes menus: Pilares, Seções, Relatórios e Configuração (fig. 6.1).

Escola de Engenharia de São Carlos -- USP Eng. Rivaldo Campos	DIMENSIONAMENTO DE PILARES EM CONCRETO ARMADO		Cartela 12/10/95 01:42:55
PILARES	SBCOBS	RELATORIOS	CONFIGURACAO

fig. 6.1. Tela principal do programa DP

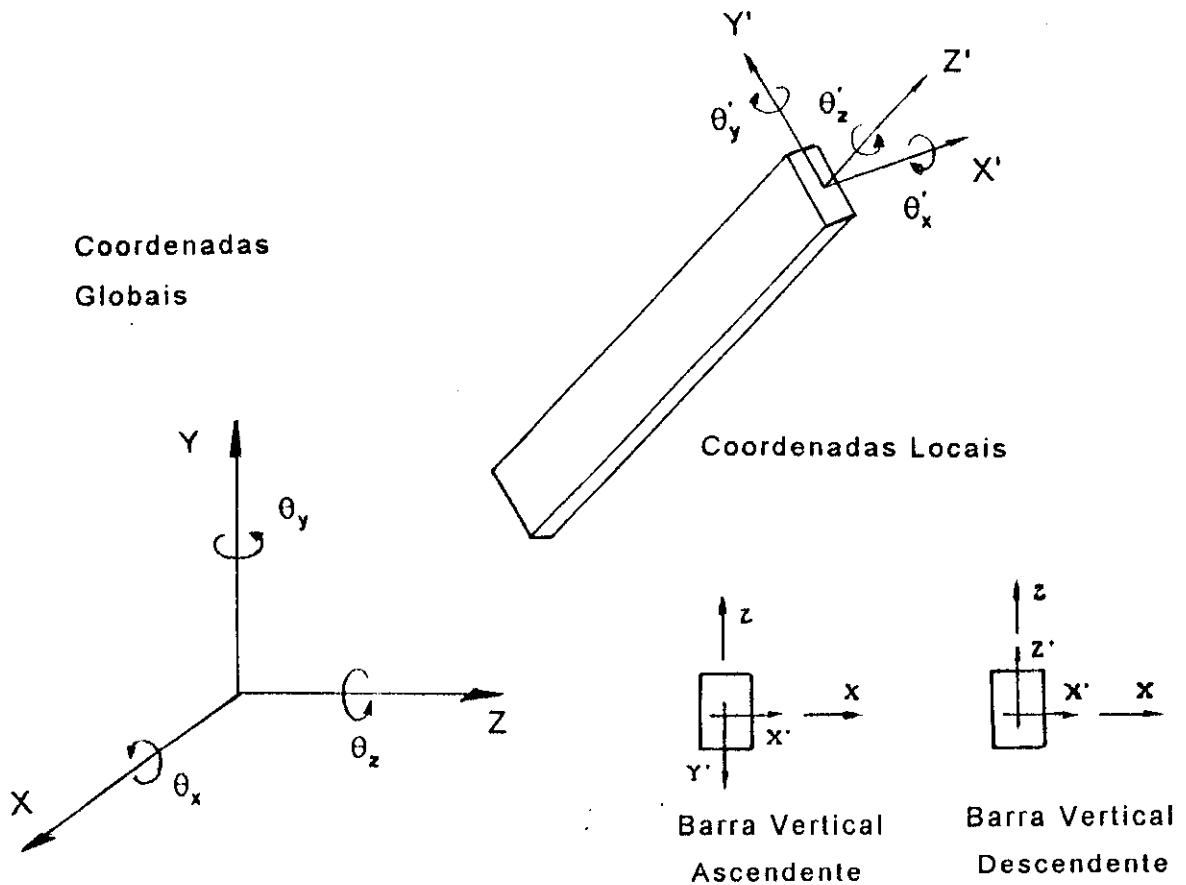


fig. 6.2. Sistemas de coordenadas global e local

Utilizando-se corretamente estes menus, consegue-se fornecer ao programa as informações necessárias para a definição dos pilares e de seus carregamentos, bem como executar seu dimensionamento. A definição do pilar, seus carregamentos e o resultado do dimensionamento são automaticamente armazenados em disco para utilização posterior. Os sistemas de coordenadas Global (SGL) e Local (SLC) adotados são apresentados na figura 6.2.

6.2.1. MENU PILARES

Através deste menu consegue-se criar ou editar dados relativos aos pilares e aos carregamentos a que ele está sujeito, executar seu dimensionamento e visualizar na tela os resultados obtidos.

Caracteriza-se um pilar pelo seu código, o qual é composto por dois dígitos, sendo este complementado por uma descrição opcional. Não é possível definir-se nenhuma característica de um pilar sem que seu código tenha sido fornecido anteriormente através da opção "Código". Esta opção permite também que seja visualizada uma lista dos pilares arquivados em disco, no diretório de trabalho.

Utilizando-se a opção "Nós", podem ser criados ou editados os pontos de intercessão das barras que constituem o pilar, sendo que para uma determinada barra admite-se que suas seções transversais e respectivas armaduras não variem. Estes pontos, ou nós, ficam definidos por suas coordenadas X, Y e Z. Nesta mesma opção pode-se caracterizar um nó como sendo um apoio do pilar. Para isso, deve-se atribuir o valor zero as suas componentes de deslocamento. Deve-se salientar que para a aplicação dos processos de dimensionamento apresentados neste trabalho, não é necessária a criação de pontos auxiliares para a discretização das barras que compoem o pilar. Isto é feito automaticamente pelo programa, que se incumbem de dividir todas as barras em um número pré-definido de trechos.

Com a opção "Barras" pode-se criar ou editar as barras que compoem o pilar e estabelecer suas incidências. Deve-se criar pelo menos uma barra para cada pilar e no caso de elementos com seção variável, uma barra para cada trecho em que a seção será admitida constante. No caso de pilares em que se deseje dividi-lo em trechos com taxas diferentes de armadura, deve-se criar uma barra para cada trecho. A cada barra é associado o código de uma seção transversal definida anteriormente.

Os carregamentos que atuam nos pilares são criados ou editados através da opção "Ações". Estes carregamentos são constituídos por esforços que podem ser aplicados aos nós ou às barras. No caso de serem aplicados às barras, podem ser concentrados ou distribuídos. Para aplicar-se um esforço concentrado em uma barra, deve-se definir seu valor e o seu ponto de aplicação, medido a partir do início da barra. Tratando-se de um esforço distribuído, além do seu valor, deve-se especificar o ponto inicial e final de aplicação, referenciados ao início da barra. É necessário especificar também a combinação dos diversos carregamentos e o coeficiente de minoração ou de majoração que afetará cada um dos carregamentos.

A opção "Cálculo" possibilita o dimensionamento da estrutura, a visualização e a edição dos resultados obtidos. Para realizar o dimensionamento, é necessário especificar a combinação de carregamentos que está atuando na estrutura.

6.2.2. MENU SEÇÕES

Este menu possibilita a completa definição das seções transversais das barras que compoem o pilar. Estas seções podem apresentar qualquer forma poligonal regular ou irregular. No caso dela apresentar contorno curvilíneo, este deve ser discretizado em trechos retilíneos. A forma circular é aceita pelo programa, que automaticamente a converte em um polígono regular com número de lados pré-definido.

As seções são gerenciadas pelo programa como um conjunto de figuras. Assim, o contorno externo da seção é definido como sendo a figura de número 1. No caso da seção ser vazada, definem-se novas figuras, às quais se associam números maiores que 1, caracterizando-se desta forma os contornos dos vazios internos desta seção. Quando uma figura está sendo criada, é necessário informar se trata-se de um polígono regular ou irregular, de um retângulo ou de uma circunferência.

No caso de polígonos é solicitado o número de lados que o compoem e a quantidade preponderante de barras da armadura que se distribuem ao longo de cada lado. Se o polígono for regular, é solicitado ainda o fornecimento do seu giro no sentido anti-horário e a posição do seu CG em relação ao SLC, após o que, as coordenadas de seus vértices são geradas automaticamente. No caso de polígonos irregulares, é necessário que o próprio usuário forneça as coordenadas dos vértices.

Para figuras retangulares tudo acontece da mesma forma como para os polígonos regulares, exceto pelo fato de serem solicitadas as dimensões de sua base e de sua altura, em lugar do número de lados. Entende-se como base da figura retangular, o lado paralelo ao eixo X' no SLC.

Tratando-se de figuras circulares, é solicitado o seu diâmetro e a posição do seu CG em relação ao SLC. Além disso, deve-se informar a quantidade de barras que compoem a parte da armadura total associada a esta figura, estabelecendo-se também um giro para esta armadura em torno do CG da circunferência.

Em todas as seções, assume-se a existência de uma barra da armadura em cada vértice das figuras que as compoem. As barras da armadura tem sua posição em relação ao SLC determinadas automaticamente.

O menu "Seções" possibilita ainda a edição, eliminação ou visualização de qualquer seção, sendo possível através dele, acessar

uma lista de seções já cadastradas. A figura 6.3 reproduz a tela de visualização de seções do programa DP, onde se apresenta uma seção composta por diversos tipos de figuras e suas características geométricas. Pela figura, nota-se que o SLC não coincide com o CG da seção. Isto acontece porque na definição da seção, adotou-se o CG da figura externa como sendo a origem do SLC. Durante o dimensionamento, será aplicada uma translação ao SLC, de tal maneira que sua origem coincida com o CG da seção.

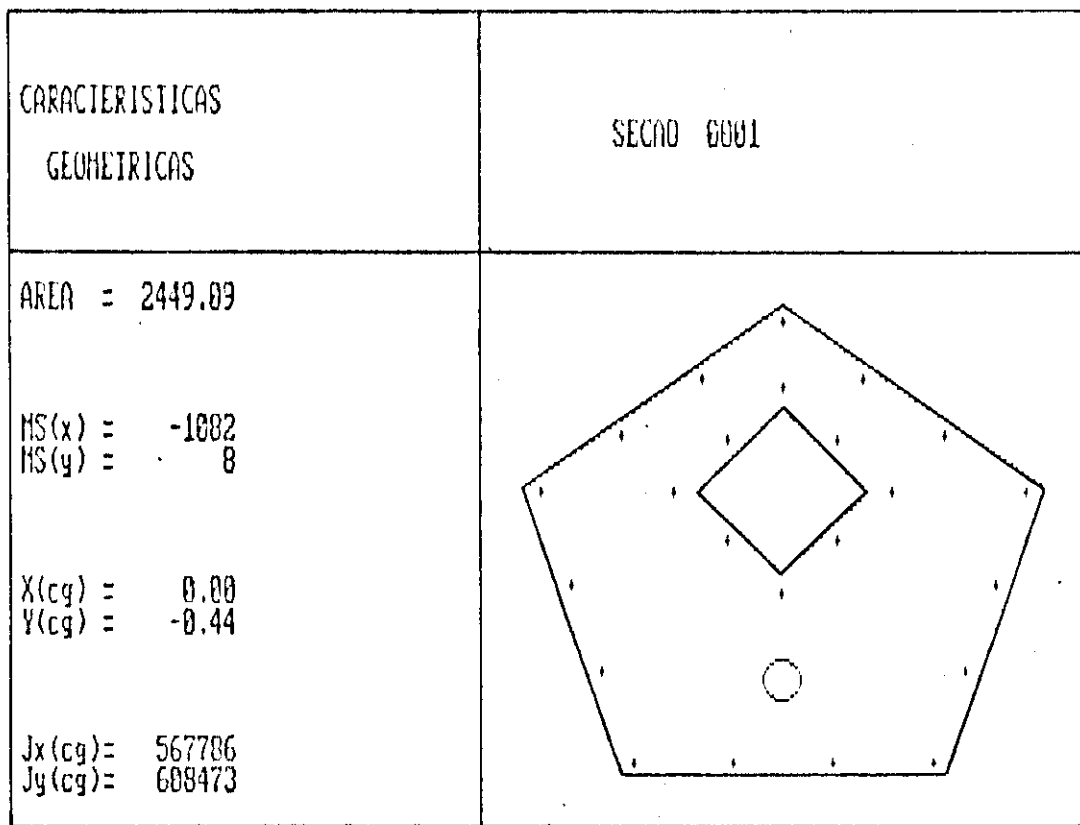


fig. 6.3. Seção transversal gerada pelo programa DP

6.2.3. MENU RELATÓRIOS

Através deste menu, pode-se imprimir todos os dados relativos a um pilar e às seções de suas barras. Também são impressos os carregamentos que atuam nela, bem como os resultados do seu dimensionamento e os parâmetros adotados.

6.2.4. MENU CONFIGURAÇÃO

Utilizando-se o menu "Configuração", consegue-se atribuir valores para todos os parâmetros adotados na definição das estruturas e no seu dimensionamento, sem que seja necessário alterar o programa. Estes parâmetros são agrupados de acordo com sua função, para simplificar o acesso a eles. Desta forma, escolhendo-se alguma dentre as opções: "Estruturas", "Seções", "Dimensionamento", "Materiais" e "Sistema", pode-se alterar qualquer parâmetro.

Através da opção "Estrutura", altera-se o número de trechos em que as barras são discretizadas para o seu dimensionamento.

Com a opção "Seções", define-se o polígono regular que discretiza as figuras circulares, a distância entre o CG das barras da armadura e a face da figura.

Na opção "Dimensionamento", especifica-se as taxas mínima e máxima da armadura.

Pode-se definir as características do concreto e do aço, através da opção "Materiais". Estabelece-se assim os tipos e as resistências do aço e do concreto, os coeficientes γ_s e γ_c de minoração destas resistências para o dimensionamento no estado limite último e o coeficiente γ_c e γ_s a serem utilizados na determinação dos deslocamentos da estrutura.

Finalmente, o drive e o diretório de trabalho são especificados a partir da opção "Sistema".

Todos as informações fornecidas ficam gravados em disco, juntamente com os dados que definem o pilar.

6.2.5. ENTRADA DE DADOS

Em todos os menus onde é solicitada a entrada de dados em série, deve-se fazer uso de teclas especiais para este fim. O programa DP sempre apresentará os dados fornecidos em ordem crescente. Assim, os nós do pilar, suas barras e os vértices das seções que as compoem serão apresentadas na tela sempre ordenados do menor para o maior, independentemente da ordem de entrada dos dados. Um recurso importante utilizado pelo programa é que os dados são exibidos em uma janela na tela e o usuário pode "navegar" por eles utilizando as teclas de movimentação. Caso se deseje eliminar algum dado, basta pressionar a tecla "delete" e este será apagado. Para se criar um novo dado, pressiona-se a tecla "insert" e será aberto um novo campo para a digitação, sendo que o número associado àquele dado é atribuído automaticamente, incrementando o valor do campo anterior.

6.2.6. UNIDADES UTILIZADAS

As dimensões da estrutura são expressas em metros, e das seções em centímetros. As unidades das resistências características do aço e do concreto são assumidas automaticamente pelo programa a partir das especificações destes materiais. A área de armadura resultante do dimensionamento é expressa em centímetros quadrados. Os esforços solicitantes são expressos em kN e kN.m .

CAPÍTULO 7

EXEMPLOS DE DIMENSIONAMENTO

Neste capítulo são apresentados alguns exemplos de dimensionamento efetuados com base nos procedimentos desenvolvidos neste trabalho e fazendo-se uso do programa "DP". A escolha dos exemplos se deu visando demonstrar a eficiência e a versatilidade do programa, permitindo, ao mesmo tempo, que se comparasse os resultados obtidos com os apresentados por outros autores.

EXEMPLO 1:

Dimensionar com armadura constante o pilar apresentado na figura 7.1. Deve-se considerar as ações como sendo de curta duração. As resistências dos materiais para a determinação dos deslocamentos de segunda ordem serão tomadas iguais às resistências de cálculo, determinadas com os coeficientes: $\gamma_c = 1,4$ e $\gamma_s = 1,15$.

Dados: Aço CA-50 *A* $\gamma_s = 1,15$ $\gamma_c = 1,4$

Concreto C21

F = 2520 kN

H = 70,5 kN

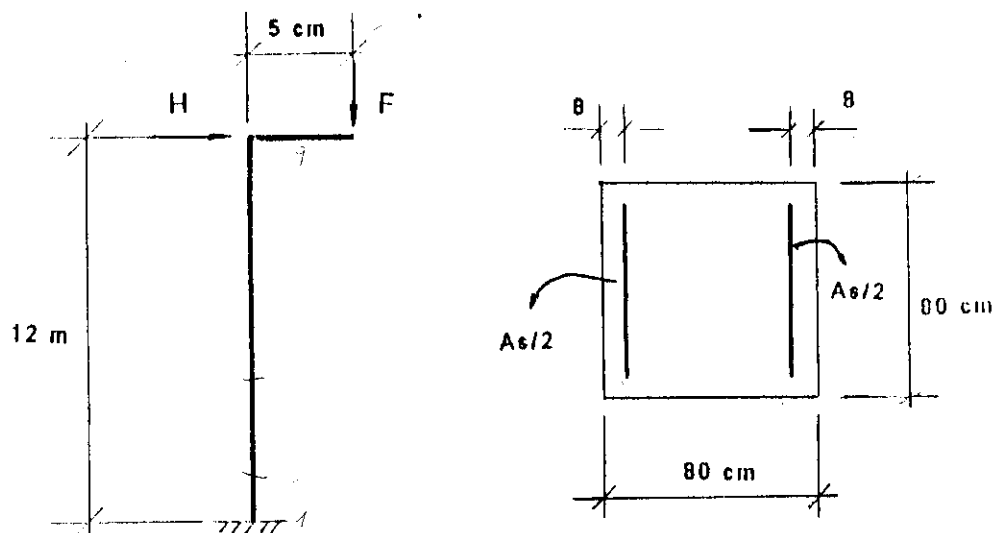


fig. 7.1 - Pilar do exemplo 1

Para aplicação do processo iterativo de dimensionamento, adota-se, neste caso, o pilar composto por uma única barra. No entanto, esta barra será discretizada automaticamente, de acordo com o especificado no menu "Configuração" do programa "DP". Definiu-se neste exemplo, que a discretização do pilar seria em 8 trechos.

Procedendo-se ao dimensionamento, obteve-se os seguintes resultados:

Área de aço: $A_s = 84,16 \text{ cm}^2$

$A_s = 21,04$

Estes valores apresentam um variação igual a 5,2% em relação aos apresentados por Paula (1988).

EXEMPLO 2:

Dimensionar o pilar retangular com armadura constante, submetido à flexão composta oblíqua apresentado na fig. 7.2. Deve-se desprezar o efeito da fluência e adotar-se tanto no dimensionamento como na determinação dos deslocamentos os coeficientes de minoração da resistência dos materiais: $\gamma_c = 1,2$ e $\gamma_s = 1,0$.

Dados: Aço CA-50A

Concreto C15

$F_k = 531,300 \text{ kN}$

$M_{xk} = 33,125 \text{ kN.m}$

$M_{zk} = 22,525 \text{ kN.m}$

$L = 3,75 \text{ m}$

$b = 25 \text{ cm}$

$h = 50 \text{ cm}$

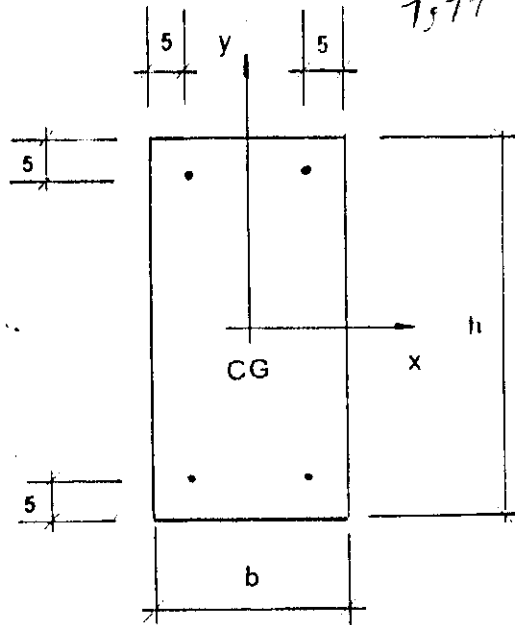
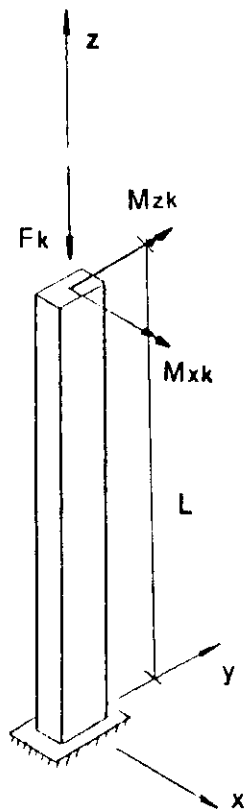


fig. 7.2 - Pilar do exemplo 2

No dimensionamento deste pilar adotou-se a discretização em 10 trechos, resultando numa armadura $A_s = 8,21 \text{ cm}^2$. Este valor diverge em cerca de 3% do valor apresentado por Marcotti (1984). Isto se deve provavelmente ao fato de terem sido obtidos os valores do carregamento utilizado neste exemplo a partir do ábaco elaborado por aquele autor para este pilar.

EXEMPLO 3:

Dimensionar o pilar em L da fig. 7.3, considerando a armadura constante em todas as seções. As cargas são de curta duração e estão aplicadas no CG da seção. Os coeficientes de minoração da resistência dos materiais são:

Em serviço: $\gamma_c = 1,2$ e $\gamma_s = 1,0$.

No estado limite último: $\gamma_c = 1,4$ e $\gamma_s = 1,15$.

Dados: Aço CA-50B

Concreto C18

Fd = 1000 kN

Hxd = 74 kN

Hyd = 18 kN

Handwritten calculations:

	30	8
1,53	30	3,9
e	30	4,95
	55	
	50	4,15

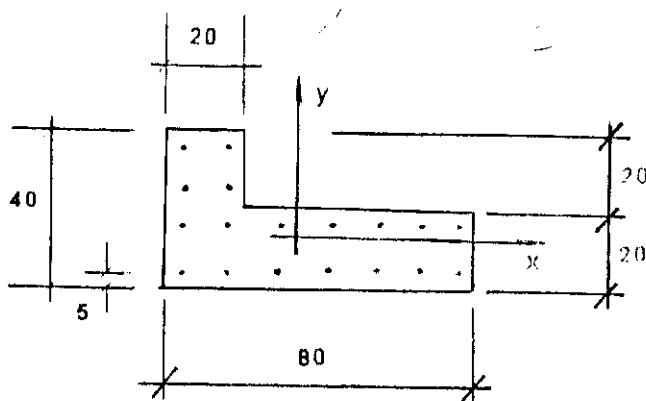
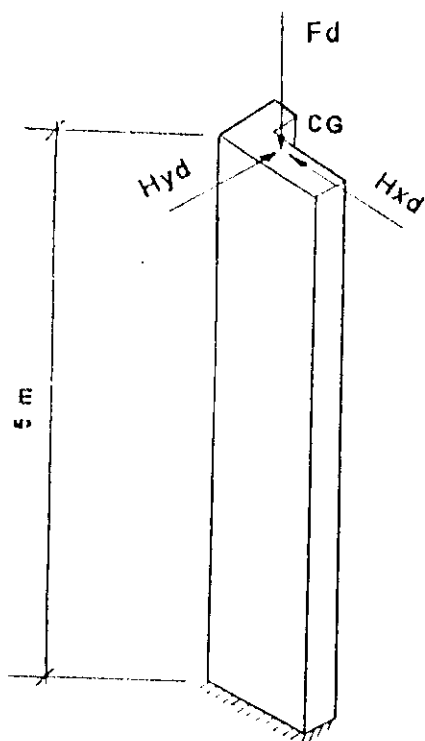


fig. 7.3 - Pilar do exemplo 3

Handwritten calculation: $A_s = 2,517$

Neste exemplo adotou-se a discretização do pilar em 4 trechos. Foi obtida a armadura total $A_s = 45,32 \text{ cm}^2$, a qual é constituída por barras com a mesma bitola. Este valor é cerca de 5% maior que o apresentado por França. Isto é devido ao fato de que no problema

original, a armadura era constituída por barras de bitolas diferentes. Além disso, os valores de cálculo usados neste exemplo foram obtidos a partir do ábaco desenvolvido por aquele autor, havendo certa imprecisão implícita neste procedimento.

CAPÍTULO 8

CONCLUSÕES

Este trabalho fornece procedimentos detalhados para o estudo de pilares com a consideração da não linearidade física e geométrica, onde é frequente a existência de flexão composta oblíqua. Estes procedimentos são especialmente úteis no dimensionamento de pilares de grande esbeltez, onde é obrigatória a utilização do Método Geral e a consideração dos efeitos da fluência, bem como quando se tem pilares de seções variáveis ou de forma irregular.

O programa para computador desenvolvido de acordo com estes procedimentos mostrou-se uma ferramenta poderosa à disposição do projetista de estruturas ou do pesquisador, para enfrentar problemas que até agora só podiam ser resolvidos com métodos aproximados.

A grande versatilidade na definição dos pilares e a possibilidade de se criar automaticamente qualquer forma de seção transversal e com qualquer disposição de armadura faz deste programa o meio ideal para o estudo de pilares submetidos à flexão composta oblíqua.

Facilmente pode-se estudar o comportamento de pilares muito comuns na prática, como os pilares em L, Z ou T e para os quais não se tinha meios para dar um tratamento adequado à complexidade do problema.

Os pilares de pontes também podem ser objeto de estudo através deste programa, o qual torna viável a comparação do

desempenho de diferentes concepções estruturais, viabilizando a redução de custos.

Apresenta também interesse a comparação de desempenho entre elementos estruturais de seção cheia e vazada, estabelecendo-se critérios que determinem as situações em que cada um represente a melhor solução.

Enfim, o programa apresentado neste trabalho é particularmente útil tanto para aplicação imediata na resolução de problemas usuais da engenharia de estruturas, como no desenvolvimento de pesquisas que somente são viáveis dispondo-se de um meio preciso, geral e eficiente para a análise dos fenômenos em estudo.

Sugere-se como continuidade deste trabalho, a ampliação das hipóteses adotadas, para, por exemplo, levar-se em conta o efeito da torção, ou desenvolver-se os procedimentos aqui apresentados, para materiais com comportamento distinto, como é o caso dos concretos de alto desempenho.

REFERÊNCIAS BIBLIOGRÁFICAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS - NBR 6118:
Procedimentos para o projeto e execução de obras de concreto
armado. Rio de Janeiro, 1986.

----- NBR-6120: Cargas para o cálculo de estruturas de
edificações. Rio de Janeiro, 1980.

----- NBR-6123: Forças devidas ao vento em edificações. Rio de
Janeiro, 1988.

----- NBR-7480: Barras e fios de aço destinados a armaduras para
concreto armado. Rio de Janeiro, 1985.

----- NBR-8681: Ações e segurança nas estruturas. Rio de
Janeiro, 1984

AUFIERO, L. - Estabilidade de colunas isostáticas de concreto armado.
São Carlos, 1977. Escola de Engenharia de São Carlos - USP.
Dissertação (Mestrado em Engenharia).

BACARJI, B. - Análise de estruturas de edifícios: Projeto de pilares. São
Carlos, 1993. Escola de Engenharia de São Carlos - USP.
Dissertação (Mestrado em Engenharia).

- BORTOLIN, A. A. - Flexão reta ou oblíqua em peças de concreto armado. São Paulo: Fapesp, 1988. (Relatório). BORTOLIN, A. A. - Sobre a racionalização e a automação do cálculo de edifícios de concreto armado. São Carlos, 1991. Escola de Engenharia de São Carlos - USP. Dissertação (Mestrado em Engenharia).
- BUCHAIN, R. - Efeitos de segunda ordem e estado limite último de instabilidade em pilares de concreto. São Paulo, 1979. USP. Dissertação (Mestrado em Engenharia)
- BURKE JÚNIOR, J. U. - Flexão normal composta: peças de seção retangular em concreto armado com armadura lateral, tabelas para dimensionamento. São Paulo, 1974. Departamento de Livros e Publicações do Grêmio Politécnico.
- CANDREVA, P. - Considerações sobre o equilíbrio e compatibilidade estrutural. São Paulo, 1981. Editora do Grêmio Politécnico.
- CEB/FIP - Comité Euro-Internacional du Béton - Manual of buckling and instability. V. 123. Paris, 1978.
- FORNI, F. E. B. - Flexão Oblíqua Composta no Concreto Armado: Seções Quaisquer. São Paulo, 1980. USP. Dissertação (Mestrado em Engenharia).
- FRANÇA, R. L. S. - Relações Momento-Curvatura em Peças de Concreto Armado Submetidas à Flexão Oblíqua Composta. São Paulo, 1984. USP. Dissertação (Mestrado em Engenharia)
- FUSCO, P. B. - Estruturas de Concreto: Solicitações Normais. Rio de Janeiro, 1981. Guanabara Dois.
- HEHL, M. E. - Linguagem de programação estruturada: FORTRAN 77. São Paulo, 1986. McGraw-Hill.

- MARCOTTI, P. - Instabilidade na Flexão Composta Oblíqua de Pilares de Concreto Armado. São Paulo, 1984. USP. Dissertação (Mestrado em Engenharia).
- MENDES NETO, F. - Estudo de pilares de concreto armado submetidos à flexão oblíqua composta. São Paulo, 1991. USP. Dissertação (Mestrado em Engenharia).
- MOSHER, F. E.; SCHNEIDER, D. I. - Turbo Basic: Guia do usuário. São Paulo, 1990. McGraw-Hill
- PAULA, J. A. - Algoritmos para o Estudo de Pilares Esbeltos em Concreto Armado Sujeitas a Solicitações Normais. São Carlos, 1989. Escola de Engenharia de São Carlos - USP. Dissertação (Mestrado em Engenharia).
- PINHEIRO, L. M.; GIONGO, J. S. - Concreto Armado: Propriedades dos Materiais. São Carlos, 1986. Escola de Engenharia de São Carlos - USP. Apostila.
- PINHEIRO, L. M. - Concreto Armado: Tabelas e ábacos. São Carlos, 1986. Escola de Engenharia de São Carlos - USP. Apostila.
- RAMALHO, J. A. A. - Clipper 5.0: Guia de referência do programador. São Paulo, 1990. Makron, McGraw-Hill.
- RODRIGUES, R. O. - Automatização do Projeto Estrutural de Pilares de Concreto Armado. São Carlos, 1992. Escola de Engenharia de São Carlos - USP. Dissertação (Mestrado em Engenharia).
- SANTOS, L. M. - Cálculo de concreto armado segundo a NB-1/76 e o CEB (Volumes I e II). São Paulo, 1981. Editora LMS.

- SAUAIA, W. - Estudo comparativo dos processos aproximados de dimensionamento à flexão oblíqua composta com redução a duas flexões normais. São Paulo, 1981. USP. Dissertação (Mestrado em Engenharia).
- SILVA, R. M. - Análise de estruturas tridimensionais de edifícios altos com núcleos resistentes considerando o efeito P-DELTA. São Carlos, 1989. Escola de Engenharia de São Carlos - USP. Dissertação (Mestrado em Engenharia).
- VENTURINI, W. S. - Dimensionamento de peças retangulares de concreto armado solicitadas à flexão reta. São Carlos, 1989. Escola de Engenharia de São Carlos - USP.
- VENTURINI, W. S.; BORTOLIN, A. A. - Dimensionamento de peças retangulares de concreto armado solicitadas à flexão oblíqua. São Carlos, 1992. Escola de Engenharia de São Carlos - USP.

APENDICE

* DP.PRG - MODULO PRINCIPAL DO SISTEMA DP

CLEAR

* CONFIGURAÇÃO DO AMBIENTE

```
SET ESCAPE ON
SET CURSOR OFF
SET CONFIRM OFF
SET TALK OFF
SET STATUS OFF
SET DATE BRIT
SET SCORE OFF
SET DELETE ON
SET SAFE OFF
SET WRAP ON
SET DECI TO 3
SET FIXED OFF
READEXIT(T.)
```

* DEFINIÇÃO DO ARQUIVO DE ROTINAS ATIVO

```
SET PROCEDURE TO DP_ROT
```

* CONSTRUÇÃO DA TELA

```
COR1 = "W/R" && BRANCO/VERMELHO      - TELA
COR2 = "R/W" && VERMELHO/BRANCO      - TELA REVERSA
COR3 = "N/R" && PRETO/VERMELHO      - SOMBRA
COR4 = "W/N" && BRANCO/PRETO
COR5 = "W/G" && BRANCO/VERDE      - MENUS
COR6 = "G/W*" && VERDE/BRANCO PULSANTE - AVISOS
```

DO DP_TELA

* DEFINIÇÃO DO DIRETÓRIO DE TRABALHO

```
IF .NOT. FILE("DIRETORI.MEM")
    DRT1 = ''
    SAVE TO DIRETORI.MEM ALL LIKE DRT1
ENDIF
```

```
RESTORE FROM DIRETORI ADDITIVE
```

* DEFINIÇÃO DAS VARIÁVEIS DO SISTEMA

```
NCOL=132
IA5=CHR(14)
ID5=CHR(20)
I10=CHR(30)+"0"
I12=CHR(30)+"2"
IA20=CHR(15)
IA18=CHR(18)
IE18=CHR(27)+"0"
IE16=CHR(27)+"2"
```

```
ESTRU=SPAC(2)
DESC_ATI='NENHUMA'
```

```
SIN = 0
COS = 0
TAN = 0
```

```
NP1=0
CARGI = ''
```

```
DECLARE VETOR1[10],VETOR3[10]
```

```
OP=1
```

* ABRE OU CRIA OS ARQUIVOS DE DADOS

```
DO DP_INDE
DO SUJA
```

```
DO AVISO WITH 23,'PILAR ATIVO: '+ESTRU+' - '+ALLTRIM(DESC_ATI)
```

```

SET COLOR TO &COR1

* APRESENTAÇÃO DO MENU PRINCIPAL

DO WHILE .T.

    @ 05,08 PROMPT " PILARES "
    @ 05,26 PROMPT " SECOES "
    @ 05,40 PROMPT " RELATORIOS "
    @ 05,58 PROMPT " CONFIGURACAO "

    MENU TO OP
    DPI=OP

    DO CASE

        CASE OP = 0
            CLOSE ALL
            SET CURSOR ON
            SET COLOR TO &COR1
            CLEAR
            RETURN
            * SE FOR PRESSIONADO <ESC> FINALIZA

        CASE OP = 1
            DO DP_EST

        CASE OP = 2 .AND. VAL(ESTRU)>0
            DO DP_SEC

        CASE OP = 3 .AND. VAL(ESTRU)>0
            DO DP_REL

        CASE OP = 4
            DO DP_CONF

    ENDCASE

DO SUJA
SET COLOR TO &COR1
@ 23,01 CLEAR TO 23,78
DO AVISO WITH 23,"PILAR ATIVO: "+ESTRU+" - "+ALLTRIM(DESC_ATI)

OP=DPI
ENDDO
RETURN

```

***DP_EST.PRG - GERENCIAMENTO DE PILARES EM CONCRETO ARMADO**

```

SELE 10
EST0 = 1

DECLARE TPI[5]
TPI[1] = " CODIGO "
TPI[2] = " NOS "
TPI[3] = " BARRAS "
TPI[4] = " ACOES "
TPI[5] = " CALCULO "

DO WHILE .T.

    X1=09
    X2=13
    Y1=09
    Y2=17

    DO MENU WITH 'TPI','EST0'

    DO CASE

        CASE OP = 0
            EXIT
            * SE FOR PRESSIONADO <ESC> FINALIZA

        CASE OP = 1
            DO DP_EST1
            * SELEÇÃO DE PILARES

        CASE OP = 2 .AND. VAL(ESTRU)>0
            DO DP_EST2
            * ENTRADA DE DADOS DOS NOS

        CASE OP = 3 .AND. VAL(ESTRU)>0
            DO DP_EST3
            * ENTRADA DOS DADOS DAS BARRAS

        CASE OP = 4 .AND. VAL(ESTRU)>0
            DO DP_EST4
            * ENTRADA DAS CARGAS APLICADAS NO PILAR

```

```

CASE OP = 5 .AND. VAL(ESTRU)>0
DO DP_EST5
  * DIMENSIONAMENTO DO PILAR

ENDCASE

DO SUJA
SET COLOR TO &COR1

DO AVISO WITH 23,'PILAR ATIVO: '+ALLTRIM(ESTRU)+' - '+ALLTRIM(DESC_ATI)

SELE 10
ENDDO
RETURN

```

* DP_EST1.PRG - MÓDULO DE SELEÇÃO DE PILARES

```

EST1= 1
SELE 10

GO TOP

KEYBOARD CHR(1)
DECLARE VETOR1[2],VETOR3[2]

VETOR3[1] = "PILAR "
VETOR3[2] = "DESCRICA0"

VETOR1[1] = "EST"
VETOR1[2] = "DESCRICA0"

DBEDIT(07,1,21,78,VETOR1,"FUNCAO2",.T.,VETOR3)

RETURN

```

* DP_EST2.PRG - MÓDULO PARA CADASTRAMENTO DOS NÓS

```

EST2=1
SET CURSOR ON

DECLARE TP1[3]
TP1[1] = " CRIAR "
TP1[2] = " EDITAR "
TP1[3] = " APOIOS "

DO WHILE .T.

  X1=13
  X2=15
  Y1=15
  Y2=22

  DO MENU WITH 'TP1','EST2'

  TELA1 = SAVESCREEN(7,0,24,79)

  DO CASE

    CASE OP = 0
      EXIT
      * SE FOR PRESSIONADO <ESC> FINALIZA

    CASE OP = 1
      DO DP_EST21
      * CRIACAO DE NOS

    CASE OP = 2
      DO DP_EST22
      * ALTERACAO DE NOS

    CASE OP = 3
      DO DP_EST23
      * DEFINICAO DOS APOIOS

  ENDCASE

  RESTSCREEN(7,0,24,79,TELA1)

ENDDO
RETURN

```

* DP_EST21.PRG - GERAÇÃO AUTOMÁTICA DE NÓS

```
SET CURSOR ON

X1=11
X2=17
Y1=18
Y2=71

DO SUJA
DO QUADRO
DO AVISO WITH 12,'DEFINICAO DA QUANTIDADE DE NOS'
@ 15,18 SAY ' QUANTIDADE DE NOS DO PILAR:' GET NP1 PICT '999'
READ
SET COLOR TO &COR1

IF NP1=0 .OR. LASTKEY()=27
  SET CURSOR OFF
  RETURN
ENDIF

SELE 5

DELE ALL

I=1

DO WHILE I <= NP1
  APPEND BLANK
  REPLA EST WITH ESTRU,NO WITH STRZERO(1,3)
  I = I+1
ENDDO

GO TOP
PT=4

DECLARE VETOR4[4],VETOR5[4]

VETOR5[1] = "NUMERO DO NO"
VETOR5[2] = "COORDENADA X (M)"
VETOR5[3] = "COORDENADA Y (M)"
VETOR5[4] = "COORDENADA Z (M)"

VETOR4[1] = "NO"
VETOR4[2] = "X"
VETOR4[3] = "Y"
VETOR4[4] = "Z"

KEYBOARD CHR(1)
DBEDIT(07,1,21,78,VETOR4,"FUNCAO3"..T.,VETOR5)

SET CURSOR OFF
```

* DP_EST22.PRG - EDIÇÃO DAS COORDENADAS DOS NÓS

```
SET CURSOR ON
SELE 5
GO TOP
PT=4
DECLARE VETOR4[4],VETOR5[4]

VETOR5[1] = "NUMERO DO NO (M)"
VETOR5[2] = "COORDENADA X (M)"
VETOR5[3] = "COORDENADA Y (M)"
VETOR5[4] = "COORDENADA Z (M)"

VETOR4[1] = "NO"
VETOR4[2] = "X"
VETOR4[3] = "Y"
VETOR4[4] = "Z"

KEYBOARD CHR(4)
DBEDIT(07,1,21,78,VETOR4,"FUNCAO3"..T.,VETOR5)

SET CURSOR OFF
```

* DP_EST23.PRG - DEFINIÇÃO DOS APOIOS E RECALQUES

```
SET CURSOR ON
SELE 6

PT=3
KEYBOARD CHR(1)

DECLARE VETOR4[3],VETOR5[3],VETOR6[3]

VETOR5[1] = "NUMERO DO NO"
VETOR5[2] = "DIRECAO (DX,DY,DZ,GX,GY,GZ)"
VETOR5[3] = "DESLOCAMENTO (M,RAD)"
```

```

VETOR4{1} = "NO"
VETOR4{2} = "DR"
VETOR4{3} = "DESLO"

VETOR6{1} = "999"
VETOR6{2} = "@!A"
VETOR6{3} = "@K"

DBEDIT(07,1,21,78,VETOR4,"FUNCAO3",VETOR6,VETOR5)

SET CURSOR OFF

```

* DP_EST3.PRG - MÓDULO PARA CADASTRAMENTO DAS BARRAS

```

EST3=1

DECLARE TPI{2}
TPI{1} = " CRIAR "
TPI{2} = " EDITAR "

DO WHILE .T.

    X1=14
    X2=15
    Y1=15
    Y2=22
    DO MENU WITH 'TPI','EST3'
    TELA1=SAVESCREEN(7,0,24,79)

    DO CASE

        CASE OP = 0
            EXIT
            * SE FOR PRESSIONADO <ESC> FINALIZA

        CASE OP = 1
            DO DP_EST31
            * CRIACAO DE BARRAS

        CASE OP = 2
            DO DP_EST32
            * ALTERACAO DE BARRAS

    ENDCASE
    RESTSCREEN(7,0,24,79,TELA1)

ENDDO
RETURN

```

* DP_EST31.PRG - GERAÇÃO AUTOMÁTICA DE BARRAS

```

SET CURSOR ON
NB1=0

X1=11
X2=17
Y1=18
Y2=71

DO SUJA
DO QUADRO

DO AVISO WITH 12,'DEFINICAO DA QUANTIDADE DE BARRAS'
@ 15,18 SAY ' QUANTIDADE DE BARRAS DO PILAR:' GET NB1 PICT '999'
READ

IF NB1=0
    SET CURSOR OFF
    RETURN
ENDIF

SET COLOR TO &COR1

SELE 7
DELE ALL

I=1

DO WHILE I <= NB1
    APPEND BLANK
    REPLA EST WITH ESTRU,BAR WITH STRZERO(1,3)
    I = I+1
ENDDO

GO TOP
PT=4

DECLARE VETOR4{4},VETOR5{4}

```

```

VETOR5[1] = "NUMERO DA BARRA"
VETOR5[2] = "NO' INICIAL"
VETOR5[3] = "NO' FINAL"
VETOR5[4] = "SECAO"

VETOR4[1] = "BAR"
VETOR4[2] = "NI"
VETOR4[3] = "NF"
VETOR4[4] = "SECAO"

KEYBOARD CHR(4)
DBEDIT(07,1,21,78,VETOR4,"FUNCAO3",.T.,VETOR5)

SET CURSOR OFF

```

* DP_EST32.PRG - EDIÇÃO DOS DADOS DAS BARRAS

```

SET CURSOR ON
SELE 7
GO TOP
PT=4
DECLARE VETOR4[4],VETOR5[4]

VETOR5[1] = "NUMERO DA BARRA"
VETOR5[2] = "NO' INICIAL"
VETOR5[3] = "NO' FINAL"
VETOR5[4] = "SECAO"

VETOR4[1] = "BAR"
VETOR4[2] = "NI"
VETOR4[3] = "NF"
VETOR4[4] = "SECAO"

KEYBOARD CHR(1)
DBEDIT(07,1,21,78,VETOR4,"FUNCAO3",.T.,VETOR5)

SET CURSOR OFF

```

* DP_EST4.PRG - GERAÇÃO DOS CARREGAMENTOS

```

EST4=1
SELE 11

DECLARE TP2[2]

TP2[1] = " CARREGAMENTO "
TP2[2] = " COMBINAÇÕES "

DO WHILE .T.

    X1=14
    X2=15
    Y1=15
    Y2=28

    DO MENU WITH 'TP2',EST4

    TELA1 = SAVESCREEN(7,1,23,78,)

    DO CASE

        CASE OP = 0
            SELE 10
            EXIT
            * SE FOR PRESSIONADO <ESC> FINALIZA

        CASE OP = 1
            * CRIA UM NOVO CARREGAMENTO
            SELE 11

            KEYBOARD CHR(1)

            DECLARE VETOR1[2],VETOR3[2]

            VETOR3[1] = "CARREGAMENTO "
            VETOR3[2] = "DESCRICAO"

            VETOR1[1] = "CARGA"
            VETOR1[2] = "DESCRICAO"

            DBEDIT(07,1,21,78,VETOR1,"FUNCAO2",.T.,VETOR3)

        CASE OP = 2
            SELE 12
            KEYBOARD CHR(1)

            DECLARE VETOR1[2],VETOR3[2]

            VETOR3[1] = "COMBINAÇÃO "
            VETOR3[2] = "DESCRICAO "

```

```

    VETOR1[1] = "COMB"
    VETOR1[2] = "DESCRICAO"

    DBEDIT(07,1,21,78,VETOR1,"FUNCAO2",.T.,VETOR3)

    ENDCASE
    RESTSCREEN(7,1,23,78,TELA1)

    ENDDO
    RETURN

```

* DP_EST41.PRG - EDIÇÃO DAS CARGAS

```

EST41=1

X1=13
X2=14
Y1=23
Y2=39

DECLARE TP3[2]

TP3[1] = " CARGAS - NOS "
TP3[2] = " CARGAS - BARRAS "

DO WHILE .T.

    DO MENU WITH "TP3",EST41

    TELA3 = SAVESCREEN(7,1,23,78,)

    DO CASE

        CASE OP=0
            SELE 11
            EXIT
            * SE FOR PRESSIONADO <ESC> FINALIZA

        CASE OP = 1
            DO DP_EST42.PRG
            * EDITA CARGAS APLICADAS AOS NOS

        CASE OP = 2
            DO DP_EST43.PRG
            * EDITA CARGAS APLICADAS AS BARRAS

    ENDCASE
    RESTSCREEN(7,1,23,78,TELA3)

ENDDO

```

* DP_EST42.PRG - CARGAS APLICADAS AOS NOS

```

SELE 8
SET FILTER TO EST=ESTRU .AND. CARGA=CARGA1
GO TOP
PT=3
KEYBOARD CHR(1)

DECLARE VETOR4[3],VETOR5[3],VETOR6[3]

VETOR5[1] = "NO"
VETOR5[2] = "CARGAS (PX,PY,PZ,MX,MY,MZ)"
VETOR5[3] = "INTENSIDADE"

VETOR4[1] = "NO"
VETOR4[2] = "DR"
VETOR4[3] = "INTEN"

VETOR6[1] = "999"
VETOR6[2] = "@1A"
VETOR6[3] = "@K"

DBEDIT(07,1,21,78,VETOR4,"FUNCAO3",VETOR6,VETOR5)

```

* DP_EST43.PRG - CARGAS NAS BARRAS

```

SELE 9
SET FILTER TO EST=ESTRU .AND. CARGA=CARGA1
GO TOP
PT=5
KEYBOARD CHR(1)

```

```

DECLARE VETOR4[5],VETOR5[5],VETOR6[5]

VETOR5[1] = " BARRA "
VETOR5[2] = " CARGAS (PX,PY)"
VETOR5[3] = " INICIO "
VETOR5[4] = " TERMINO "
VETOR5[5] = " INTENSIDADE "

VETOR4[1] = "BAR"
VETOR4[2] = "DR"
VETOR4[3] = "INI"
VETOR4[4] = "FIM"
VETOR4[5] = "INTEN"

VETOR6[1] = "999"
VETOR6[2] = "@1A"
VETOR6[3] = "@K"
VETOR6[4] = "@K"
VETOR6[5] = "@K"

DBEDIT(07,1,21,78,VETOR4,"FUNCAO3",VETOR6,VETOR5)

```

* DP_EST5.PRG - DIMENSIONAMENTO DE PILARES

```

EST5-1
DECLARE TPI[4]

TPI[1] = " DIMENSIONAMENTO "
TPI[2] = " DESLOCAMENTOS "
TPI[3] = " ESFORCOS "
TPI[4] = " ARMADURAS "

DO WHILE .T.

    XI=11
    X2=14
    Y1=15
    Y2=31

    DO MENU WITH 'TPI','EST5'
    TELA1=SAVESCREEN(0,0,24,79)

    IF OP>1
        SELE 10
        IF EST#ESTRU
            SEEK ESTRU
        ENDIF
        SET CURSOR ON
    ENDIF

    DO CASE

        CASE OP = 0
            EXIT
            * SE FOR PRESSIONADO <ESC> FINALIZA

        CASE OP = 1
            DO DP_EST51
            * DIMENSIONAMENTO DO PILAR

        CASE OP = 2
            REPLA DESLO WITH MEMOEDIT(DESLO,7,1,21,78)
            * EDIÇÃO DOS DESLOCAMENTOS DO PILAR

        CASE OP = 3
            REPLA ESFORCOS WITH MEMOEDIT(ESFORCOS,7,1,21,78)
            * EDIÇÃO DOS ESFORÇOS APLICADAS AO PILAR

        CASE OP = 4
            REPLA ARMADURAS WITH MEMOEDIT(ARMADURAS,7,1,21,78)
            * EDIÇÃO DOS DESLOCAMENTOS DAS ARMADURAS

    ENDCASE

    SET CURSOR OFF
    RESTSCREEN(0,0,24,79,TELA1)

ENDDO
RETURN

```

* DP_EST51.PRG

```

SELE 12
TELA2=SAVESCREEN(7,1,21,78)
DRT=0

```



```

DO WHILE .T.
  X1=15
  X2=15
  Y1=23
  Y2=57
  COMBI=' '
  DO QUADRO
  SET CURSOR ON
  @ X1,Y1 SAY ' CODIGO DA COMBINACAO DE CARGAS:' GET COMBI PICT '99'
  READ
  SET CURSOR OFF

  IF VAL(COMBI) = 0 .OR. LASTKEY()=27
    RETURN
  ENDIF

  COMBI=STRZERO(VAL(COMBI),2)

  SEEK ESTRU+COMBI

  IF EOF()
    LOOP
  ENDIF

  X1=10
  X2=16
  Y1=08
  Y2=69

  DO SUJA
  DO QUADRO

  DO AVISO WITH 11, " PROCESSANDO DIMENSIONAMENTO "

  @ 13,08 SAY " PILAR: "+ESTRU+ " - "+DP_EST->DESCRICAO
  @ 15,08 SAY " ACOES: "+COMBI+ " - "+DP_COMB->DESCRICAO

  SELE 5

  GO TOP

  COPY FIELDS NO,X,Y,Z TO "DP_NOS.TXT" DELI WITH BLANK
  GO TOP

  SELE 7

  SET FILTER TO EST = ESTRU
  GO TOP

  SET ALTE TO DP_BAR.TXT
  SET CONSOLE OFF
  SET ALTE ON

  DO WHILE .NOT. EOF()
    TIPO1=""+TIPO+""
    ? BAR,NI,NF,(DP_GEO->AREA)*.0001,(DP_GEO->JX)*10**(-8),(DP_GEO->JY)*10**(-8),SECAO,TIPO1
    SKIP
  ENDDO

  ? ' '
  SET ALTE OFF
  SET ALTE TO
  SET CONSOLE ON
  GO TOP

  SELE 6
  SET FILTER TO EST=ESTRU
  GO TOP

  SET ALTE TO DP_APOIO.TXT
  SET CONSOLE OFF
  SET ALTE ON

  DO WHILE .NOT. EOF()
    DO CASE
      CASE DR='DX'
        DRT=3
      CASE DR='DY'
        DRT=2
      CASE DR='DZ'
        DRT=1
      CASE DR='GX'
        DRT=6
      CASE DR='GY'
        DRT=5
      CASE DR='GZ'
        DRT=4
    ENDCASE
    ? NO,DRT,DESLO
    SKIP
  ENDDO

  ? ' '
  SET ALTE OFF
  SET ALTE TO
  SET CONSOLE ON

```

```

GO TOP

SELE 13
SET FILTER TO EST=ESTRU .AND. COMB=COMBI
GO TOP

SET ALTE TO DP_CN.TXT
SET CONSOLE OFF
SET ALTE ON

DO WHILE .NOT. EOF()

    CARGAI=CARGA
    COEF1= GAMA_F

    SELE 8
    SET FILTER TO EST=ESTRU .AND. CARGA=CARGAI
    GO TOP

    DO WHILE .NOT. EOF()
        DO CASE
            CASE DR='PX'
                DRT=3
            CASE DR='PY'
                DRT=2
            CASE DR='PZ'
                DRT=1
            CASE DR='MX'
                DRT=6
            CASE DR='MY'
                DRT=5
            CASE DR='MZ'
                DRT=4
        ENDCASE
        ? NO,DRT,INTEN*COEF1
        SKIP
    ENDDO

    SELE 13
    SKIP

ENDDO

? ' '
SELE 8
SET FILTER TO EST=ESTRU
GO TOP
SET ALTE OFF
SET ALTE TO
SET CONSOLE ON

SELE 13
SET FILTER TO EST=ESTRU .AND. COMB=COMBI
GO TOP

SET ALTE TO DP_CB.TXT
SET CONSOLE OFF
SET ALTE ON

DO WHILE .NOT. EOF()

    CARGAI=CARGA
    COEF1=GAMA_F

    SELE 9
    SET FILTER TO EST=ESTRU .AND. CARGA=CARGAI
    GO TOP

    DO WHILE .NOT. EOF()
        DO CASE
            CASE DR='PX'
                DRT=3
            CASE DR='PY'
                DRT=2
        ENDCASE
        ? BAR,DRT,INI,FIM,INTEN*COEF1
        SKIP
    ENDDO

    SELE 13
    SKIP

ENDDO

? ' '
SET FILTER TO EST=ESTRU
GO TOP
SELE 9
SET FILTER TO EST=ESTRU
GO TOP
SET ALTE OFF
SET ALTE TO
SET CONSOLE ON

```

```

SELE 1
SET RELATION TO

SELE 2
GO TOP
DECLARE TPI{99}
SET CONSOLE OFF
SET ALTE TO DP_SEC.TXT
SET ALTE ON

SECAO1=SECAO
I=0
M=0

DO WHILE .NOT. EOF()
  I=0
  K=0
  ? SECAO1

  DO WHILE SECAO=SECAO1

    FIGURA1=FIGURA
    SELE 1
    SEEK ESTRU+SECAO1+FIGURA1
    J=0
    DO WHILE FIGURA=FIGURA1 .AND. SECAO=SECAO1
      J=J+1
      I=I+1
      SKIP
    ENDDO
    K=K+1
    TPI{K}=J
    SELE 2
    SKIP
  ENDDO

  SELE 3
  SEEK ESTRU+SECAO1
  L=0

  DO WHILE SECAO=SECAO1
    L=L+1
    SKIP
  ENDDO

  ? ' ' ,L,K
  FOR J=1 TO K
    ?? TPI{J}
  NEXT

  SELE 2
  SEEK ESTRU+SECAO1

  JJ=1
  J=0

  DO WHILE SECAO=SECAO1
    FIGURA1=FIGURA
    CIRC1=CIRC
    J=J+1
    SELE 1
    SEEK ESTRU+SECAO1+FIGURA1
    JJ=JJ
    IF CIRC1='I'
      JJ=JJ+TPI{J}-1
      INC=-1
    ELSE
      INC=1
    ENDIF

    DO WHILE FIGURA=FIGURA1 .AND. SECAO=SECAO1
      ? JJ,X/100,Y/100
      JJ=JJ+INC
      SKIP
    ENDDO
    JJ=JJ+TPI{J}
    SELE 2
    SKIP
  ENDDO

  SELE 3
  SEEK ESTRU+SECAO1
  L=0

  DO WHILE SECAO=SECAO1
    L=L+1
    ? L,X/100,Y/100
    SKIP
  ENDDO

  SELE 2
  SECAO1=SECAO
ENDDO

```

```

? SPAC(10)
? SPAC(10)
? SPAC(10)
SELE 1
SET RELA TO EST+SECAO INTO DP_SECAO
SET RELA TO EST+SECAO INTO DP_GEO ADDITIVE
SET RELA TO EST+SECAO+FIGURA INTO DP_ADDITIVE

SELE 10
SEEK ESTRU
SET ALTE TO DP_PARA.TXT
SET CONSOLE OFF
SET ALTE ON
E=18900*SQRT(VAL(FCK)*10+35)*100
?E,TRECHOS,VAL(FCK)*1000.00,GAMA_C,GAMA_CD,"CA-",LEFT(FYK,2),""+RIGHT(FYK,1)+"";
VAL(FYK)*10000.00,GAMA_S,GAMA_SD,ARM_MAX,ARM_MIN

? SPAC(10)
? SPAC(10)
? SPAC(10)
SET ALTE OFF
SET ALTE TO
SET CONSOLE ON

RUN PT WITH &DRT1

REPLACE DESLO WITH MEMOREAD('DESLO.TXT'), ESFORCOS WITH MEMOREAD('ESFORCOS.TXT'),;
ARMADURAS WITH MEMOREAD('ARMADURA.TXT')

RESTSCREEN(7,1,21,78,TELA2)
RETURN
ENDDO

```

* DP_INDE.PRG - CRIAÇÃO DOS ARQUIVOS DE DADOS DO SISTEMA DP

```

IF EMPTY(DRT1) .OR. PROCNAME(1)="DP_CONF"

CLOSE DATABASES
TITI="SELECAO DO DIRETORIO DE TRABALHO"
X1=10
X2=16
Y1=08
Y2=69
DO SUJA
DO QUADRO

DO AVISO WITH 10,TITI
DRT1=DRT1+SPAC(254-LEN(DRT1))
SET CURSOR ON
@ 14,09 SAY 'DIRETORIO:' GET DRT1 PICT '@KS49'
READ
SET CURSOR OFF
DRT1=ALLTRIM(DRT1)
ESTRU=SPAC(2)
DESC_ATI="NENHUMA"

IF PROCNAME(1)="DP_CONF"
SET COLOR TO &COR1
@ 23,01 CLEAR TO 23,78
DO AVISO WITH 23,'PILAR ATIVO: '+ESTRU+' - '+ALLTRIM(DESC_ATI)
SET COLOR TO &COR5
ENDIF

ENDIF

DRT4=DRT1

IF .NOT. EMPTY(DRT1)
DRT1=DRT1+'V'
ENDIF

DRT2=DRT1+'DP_COORD.DBF'

IF .NOT. FILE('&DRT2')
ESTRU4:={ }
AADD(ESTRU4,{'EST','C',2,0})
AADD(ESTRU4,{'SECAO','C',4,0})
AADD(ESTRU4,{'FIGURA','C',2,0})
AADD(ESTRU4,{'VERT','C',3,0})
AADD(ESTRU4,{'X','N',7,2})
AADD(ESTRU4,{'Y','N',7,2})
AADD(ESTRU4,{'NB','N',3,0})
AADD(ESTRU4,{'ESCALA','N',7,2})
DBCREATE('&DRT2',ESTRU4)
ENDIF

DRT2=DRT1+'DP_SECI.DBF'

IF .NOT. FILE('&DRT2')
ESTRU4:={ }
AADD(ESTRU4,{'EST','C',2,0})
AADD(ESTRU4,{'SECAO','C',4,0})
AADD(ESTRU4,{'FIGURA','C',2,0})

```

```

AADD(ESTRU4,('TIPO','C',4,0))
AADD(ESTRU4,('ESCALA','N',7,2))
AADD(ESTRU4,('XCG','N',7,2))
AADD(ESTRU4,('YCG','N',7,2))
AADD(ESTRU4,('GIRO','N',6,2))
AADD(ESTRU4,('CIRC','C',1,0))
DBCREATE('&DRT2',ESTRU4)
ENDIF

DRT2=DRT1+'DP_ARM.DBF'

IF .NOT. FILE('&DRT2')
ESTRU4={ }
AADD(ESTRU4,('EST','C',2,0))
AADD(ESTRU4,('SECAO','C',4,0))
AADD(ESTRU4,('FIGURA','C',2,0))
AADD(ESTRU4,('FACE','C',3,0))
AADD(ESTRU4,('BAR','C',3,0))
AADD(ESTRU4,('X','N',7,2))
AADD(ESTRU4,('Y','N',7,2))
DBCREATE('&DRT2',ESTRU4)
ENDIF

DRT2=DRT1+'DP_GEO.DBF'

IF .NOT. FILE('&DRT2')
ESTRU4={ }
AADD(ESTRU4,('EST','C',2,0))
AADD(ESTRU4,('SECAO','C',4,0))
AADD(ESTRU4,('AREA','N',10,3))
AADD(ESTRU4,('XCG','N',10,3))
AADD(ESTRU4,('YCG','N',10,3))
AADD(ESTRU4,('SX','N',14,3))
AADD(ESTRU4,('SY','N',14,3))
AADD(ESTRU4,('JX','N',14,3))
AADD(ESTRU4,('JY','N',14,3))
DBCREATE('&DRT2',ESTRU4)
ENDIF

DRT2=DRT1+'DP_NOS.DBF'

IF .NOT. FILE('&DRT2')
ESTRU4={ }
AADD(ESTRU4,('EST','C',2,0))
AADD(ESTRU4,('NO','C',3,0))
AADD(ESTRU4,('X','N',6,2))
AADD(ESTRU4,('Y','N',6,2))
AADD(ESTRU4,('Z','N',6,2))
DBCREATE('&DRT2',ESTRU4)
ENDIF

DRT2=DRT1+'DP_APOIO.DBF'

IF .NOT. FILE('&DRT2')
ESTRU4={ }
AADD(ESTRU4,('EST','C',2,0))
AADD(ESTRU4,('NO','C',3,0))
AADD(ESTRU4,('DR','C',2,0))
AADD(ESTRU4,('DESLO','N',7,5))
DBCREATE('&DRT2',ESTRU4)
ENDIF

DRT2=DRT1+'DP_BAR.DBF'

IF .NOT. FILE('&DRT2')
ESTRU4={ }
AADD(ESTRU4,('EST','C',2,0))
AADD(ESTRU4,('SECAO','C',4,0))
AADD(ESTRU4,('BAR','C',3,0))
AADD(ESTRU4,('NI','C',3,0))
AADD(ESTRU4,('NF','C',3,0))
DBCREATE('&DRT2',ESTRU4)
ENDIF

DRT2=DRT1+'DP_CN.DBF'

IF .NOT. FILE('&DRT2')
ESTRU4={ }
AADD(ESTRU4,('EST','C',2,0))
AADD(ESTRU4,('CARGA','C',2,0))
AADD(ESTRU4,('NO','C',3,0))
AADD(ESTRU4,('INTEN','N',9,3))
AADD(ESTRU4,('DR','C',2,0))
DBCREATE('&DRT2',ESTRU4)
ENDIF

DRT2=DRT1+'DP_CB.DBF'

IF .NOT. FILE('&DRT2')
ESTRU4={ }
AADD(ESTRU4,('EST','C',2,0))
AADD(ESTRU4,('CARGA','C',2,0))
AADD(ESTRU4,('BAR','C',3,0))
AADD(ESTRU4,('INI','N',6,2))
AADD(ESTRU4,('FIM','N',6,2))
AADD(ESTRU4,('INTEN','N',9,3))

```

```

        AADD(ESTRU4,('DR','C',2,0))
        DBCREATE('&DRT2',ESTRU4)
    ENDIF

DRT2=DRT1+'DP_EST.DBF'

IF .NOT. FILE('&DRT2')
    ESTRU4:={}
    AADD(ESTRU4,('EST','C',2,0))
    AADD(ESTRU4,('DESCRICAO','C',30,0))
    AADD(ESTRU4,('DESLO','M',10,0))
    AADD(ESTRU4,('ESFORCOS','M',10,0))
    AADD(ESTRU4,('ARMADURAS','M',10,0))
    AADD(ESTRU4,('TRECHOS','N',3,0))
    AADD(ESTRU4,('CIRCULO','N',3,0))
    AADD(ESTRU4,('DIST.CG','N',5,2))
    AADD(ESTRU4,('ARM_MIN','N',4,2))
    AADD(ESTRU4,('ARM_MAX','N',4,2))
    AADD(ESTRU4,('FCK','C',3,0))
    AADD(ESTRU4,('FYK','C',3,0))
    AADD(ESTRU4,('GAMA_S','N',4,2))
    AADD(ESTRU4,('GAMA_SD','N',4,2))
    AADD(ESTRU4,('GAMA_C','N',4,2))
    AADD(ESTRU4,('GAMA_CD','N',4,2))
    DBCREATE('&DRT2',ESTRU4)
ENDIF

DRT2=DRT1+'DP_COMB.DBF'

IF .NOT. FILE('&DRT2')
    ESTRU4:={}
    AADD(ESTRU4,('EST','C',2,0))
    AADD(ESTRU4,('COMB','C',2,0))
    AADD(ESTRU4,('DESCRICAO','C',30,0))
    DBCREATE('&DRT2',ESTRU4)
ENDIF

DRT2=DRT1+'DP_COMBI.DBF'

IF .NOT. FILE('&DRT2.DBF')
    ESTRU4:={}
    AADD(ESTRU4,('EST','C',2,0))
    AADD(ESTRU4,('COMB','C',2,0))
    AADD(ESTRU4,('CARGA','C',2,0))
    AADD(ESTRU4,('GAMA_F','N',4,2))
    DBCREATE('&DRT2',ESTRU4)
ENDIF

DRT2=DRT1+'DP_CARGA.DBF'

IF .NOT. FILE('&DRT2')
    ESTRU4:={}
    AADD(ESTRU4,('EST','C',2,0))
    AADD(ESTRU4,('CARGA','C',2,0))
    AADD(ESTRU4,('DESCRICAO','C',30,0))
    DBCREATE('&DRT2',ESTRU4)
ENDIF

DRT2=DRT1+'DP_SECAO.DBF'

IF .NOT. FILE('&DRT2')
    ESTRU4:={}
    AADD(ESTRU4,('EST','C',2,0))
    AADD(ESTRU4,('SECAO','C',4,0))
    AADD(ESTRU4,('DESCRICAO','C',30,0))
    DBCREATE('&DRT2',ESTRU4)
ENDIF

DRT2=DRT1+'DP_COORD.DBF'
DRT3=DRT1+'DP_COORD.NTX'

IF .NOT. FILE('&DRT3')
    USE &DRT2
    INDEX ON EST+SECAO+FIGURA+VERT TO &DRT3
ENDIF

DRT2=DRT1+'DP_SECAO.DBF'
DRT3=DRT1+'DP_SECAO.NTX'

IF .NOT. FILE('&DRT3')
    USE &DRT2
    INDEX ON EST+SECAO TO &DRT3
ENDIF

DRT2=DRT1+'DP_SECI.DBF'
DRT3=DRT1+'DP_SECI.NTX'

IF .NOT. FILE('&DRT3')
    USE &DRT2
    INDEX ON EST+SECAO+FIGURA TO &DRT3
ENDIF

DRT2=DRT1+'DP_ARM.DBF'
DRT3=DRT1+'DP_ARM.NTX'

```

```

IF .NOT. FILE("&DRT3")
  USE &DRT2
  INDEX ON EST+SECAO+FIGURA+BAR TO &DRT3
ENDIF

DRT2=DRT1+'DP_GEO.DBF'
DRT3=DRT1+'DP_GEO.NTX'

IF .NOT. FILE("&DRT3")
  USE &DRT2
  INDEX ON EST+SECAO TO &DRT3
ENDIF

DRT2=DRT1+'DP_NOS.DBF'
DRT3=DRT1+'DP_NOS.NTX'

IF .NOT. FILE("&DRT3")
  USE &DRT2
  INDEX ON EST+NO TO &DRT3
ENDIF

DRT2=DRT1+'DP_APOIO.DBF'
DRT3=DRT1+'DP_APOIO.NTX'

IF .NOT. FILE("&DRT3")
  USE &DRT2
  INDEX ON EST+NO TO &DRT3
ENDIF

DRT2=DRT1+'DP_BAR.DBF'
DRT3=DRT1+'DP_BAR.NTX'

IF .NOT. FILE("&DRT3")
  USE &DRT2
  INDEX ON EST+BAR TO &DRT3
ENDIF

DRT2=DRT1+'DP_CN.DBF'
DRT3=DRT1+'DP_CN.NTX'

IF .NOT. FILE("&DRT3")
  USE &DRT2
  INDEX ON EST+CARGA+NO TO &DRT3
ENDIF

DRT2=DRT1+'DP_CB.DBF'
DRT3=DRT1+'DP_CB.NTX'

IF .NOT. FILE("&DRT3")
  USE &DRT2
  INDEX ON EST+CARGA+BAR TO &DRT3
ENDIF

DRT2=DRT1+'DP_EST.DBF'
DRT3=DRT1+'DP_EST.NTX'

IF .NOT. FILE("&DRT3")
  USE &DRT2
  INDEX ON EST TO &DRT3
ENDIF

DRT2=DRT1+'DP_CARGA.DBF'
DRT3=DRT1+'DP_CARGA.NTX'

IF .NOT. FILE("&DRT3")
  USE &DRT2
  INDEX ON EST+CARGA TO &DRT3
ENDIF

DRT2=DRT1+'DP_COMB.DBF'
DRT3=DRT1+'DP_COMB.NTX'

IF .NOT. FILE("&DRT3")
  USE &DRT2
  INDEX ON EST+COMB TO &DRT3
ENDIF

DRT2=DRT1+'DP_COMBI.DBF'
DRT3=DRT1+'DP_COMBI.NTX'

IF .NOT. FILE("&DRT3")
  USE &DRT2
  INDEX ON EST+COMB+CARGA TO &DRT3
ENDIF

* ABERTURA DOS ARQUIVOS

SELE 10

DRT2=DRT1+'DP_EST.DBF'
DRT3=DRT1+'DP_EST.NTX'
USE &DRT2 INDEX &DRT3

```

```

SELE 11

DRT2=DRT1+'DP_CARGA.DBF'
DRT3=DRT1+'DP_CARGA.NTX'
USE &DRT2 INDEX &DRT3

SELE 14

DRT2=DRT1+'DP_SECAO.DBF'
DRT3=DRT1+'DP_SECAO.NTX'
USE &DRT2 INDEX &DRT3

SELE 4

DRT2=DRT1+'DP_GEO.DBF'
DRT3=DRT1+'DP_GEO.NTX'
USE &DRT2 INDEX &DRT3

SELE 2

DRT2=DRT1+'DP_SECI.DBF'
DRT3=DRT1+'DP_SECI.NTX'
USE &DRT2 INDEX &DRT3 ALIAS DP

SELE 1

DRT2=DRT1+'DP_COORD.DBF'
DRT3=DRT1+'DP_COORD.NTX'
USE &DRT2 INDEX &DRT3
SET RELA TO EST+SECAO INTO DP_SECAO
SET RELA TO EST+SECAO INTO DP_GEO ADDITIVE
SET RELA TO EST+SECAO+FIGURA INTO DP_ADDITIVE

SELE 3

DRT2=DRT1+'DP_ARM.DBF'
DRT3=DRT1+'DP_ARM.NTX'
USE &DRT2 INDEX &DRT3
SET RELA TO EST+SECAO+FIGURA+FACE INTO DP_COORD

SELE 5

DRT2=DRT1+'DP_NOS.DBF'
DRT3=DRT1+'DP_NOS.NTX'
USE &DRT2 INDEX &DRT3

SELE 6

DRT2=DRT1+'DP_APOIO.DBF'
DRT3=DRT1+'DP_APOIO.NTX'
USE &DRT2 INDEX &DRT3

SELE 7

DRT2=DRT1+'DP_BAR.DBF'
DRT3=DRT1+'DP_BAR.NTX'
USE &DRT2 INDEX &DRT3
DRT2=DRT1+'DP_GEO.DBF'
SET RELA TO EST+SECAO INTO DP_GEO

SELE 8

DRT2=DRT1+'DP_CN.DBF'
DRT3=DRT1+'DP_CN.NTX'
USE &DRT2 INDEX &DRT3

SELE 9

DRT2=DRT1+'DP_CB.DBF'
DRT3=DRT1+'DP_CB.NTX'
USE &DRT2 INDEX &DRT3

SELE 12

DRT2=DRT1+'DP_COMB.DBF'
DRT3=DRT1+'DP_COMB.NTX'
USE &DRT2 INDEX &DRT3

SELE 13

DRT2=DRT1+'DP_COMBI.DBF'
DRT3=DRT1+'DP_COMBI.NTX'
USE &DRT2 INDEX &DRT3
DRT2=DRT1+'DP_CARGA.DBF'
DRT3=DRT1+'DP_COMB.DBF'
SET RELATION TO EST+CARGA INTO DP_CARGA
SET RELATION TO EST+COMB INTO DP_COMB ADDITIVE

DRT1=DRT4
SAVE TO DIRETORI.MEM ALL LIKE DRT1
RETURN

```


* DP_REL.PRG - IMPRESSÃO DE RELATÓRIOS DOS PILARES

```
SELE 10
RELO = 1

DECLARE TP1[3]

TP1[1] = " PILARES "
TP1[2] = " ACOES "
TP1[3] = " RESULTADOS "

DO WHILE .T.

    X1=09
    X2=11
    Y1=41
    Y2=52

    DO MENU WITH 'TP1','RELO'

    DO CASE

        CASE OP = 0
            EXIT
            * SE FOR PRESSIONADO <ESC> FINALIZA

        CASE OP = 1 .AND. VAL(ESTRU)>0
            DO DP_REL1
            * IMPRESSÃO DAS CARACTERÍSTICAS DO PILAR

        CASE OP = 2 .AND. VAL(ESTRU)>0
            DO DP_REL2
            * IMPRESSÃO DO CARREGAMENTOS DO PILAR

        CASE OP = 3 .AND. VAL(ESTRU)>0
            DO DP_REL3
            * IMPRESSÃO DOS ESFORÇOS ATUANTES E DO DIMENSIONAMENTO DO PILAR

    ENDCASE

    DO SUJA
    SET COLOR TO &COR1
    DO AVISO WITH 23,'PILAR ATIVO: '+ALLTRIM(ESTRU)+' - '+ALLTRIM(DESC_ATI)
    SELE 10

ENDDO
RETURN
```

* DP_REL1.PRG - IMP. DAS CARACTERÍSTICAS GEOM. DOS PILARES

```
REL1= 1

DECLARE TP2[5]

TP2[1] = " NOS "
TP2[2] = " BARRAS "
TP2[3] = " SECOES-RES. "
TP2[4] = " SECOES "
TP2[5] = " APOIOS "

DO WHILE .T.

    X1=10
    X2=14
    Y1=49
    Y2=61

    DO MENU WITH 'TP2','REL1'
    TELA1 = SAVESCREEN(7,1,21,78,)
    ESTRU2 = ESTRU

    IF OP>0
        X1=10
        X2=16
        Y1=12
        Y2=65

        DO SUJA
        DO QUADRO
    ENDIF

    DO CASE
        CASE OP = 0
            RETURN
        OTHER
            DO DP_REL11
    ENDCASE

    RESTSCREEN(7,1,21,78,TELA1)

ENDDO
```

*** DP_REL11**

GO TOP
SET CONSOLE OFF

X1=10
X2=16
Y1=08
Y2=69

DO SUJA

DO CASE

CASE OP=1
TITI='RELATORIO DOS NOS'
CASE OP=2
TITI='RELATORIO DAS BARRAS'
CASE OP=3
TITI='RELATORIO RESUMIDO DAS SECOES'
CASE OP=4
TITI='RELATORIO DAS SECOES'

DO QUADRO

DO AVISO WITH 10,TITI+" DO PILAR "+ESTRU

DO WHILE .T.

DO AVISO WITH 16,'TECLE: [ENTER] LISTAR TUDO [ESC] CANCELAR'
SECAO2=SPAC(3)
SET CURSOR ON
@ 12,09 SAY "SECAO:" GET SECAO2 PICT '999'
READ
SET CURSOR OFF

IF LASTKEY()=27
RESTSCREEN(7,1,21,78,TELA1)
RETURN
ENDIF

SECAO2=STRZERO(VAL(SECAO2),4)

IF VAL(SECAO2) > 0
SELE 1
SEEK ESTRU+SECAO2

IF EOF()
TONE(261,7,2)
TONE(392,7)
DO AVISO WITH 16,SPAC(60)
SET COLOR TO &COR6
DO AVISO WITH 16,"SECAO NAO CADASTRADA !"
INKEY(2)
SET COLOR TO &COR5
LOOP

ELSE
@ 14,09 SAY 'DESCRICAO: '+DP_SECAO->DESCRICAO
DO AVISO WITH 16,SPAC(60)
DO AVISO WITH 16,"TECLE: [ENTER] PARA PROSSEGUIR [ESC] ALTERAR"
K=INKEY(0)
IF LASTKEY()=27
LOOP
ENDIF
ENDIF

ELSE
SELE 1
GO TOP
ENDIF
EXIT
ENDDO

CASE OP=5
TITI='RELATORIO DOS APOIOS'

ENDCASE

DO QUADRO

DO AVISO WITH 11,TITI+" DO PILAR "+ESTRU
DO AVISO WITH 13,"VERIFIQUE A IMPRESSORA E TECLE:"
DO AVISO WITH 15,"[ENTER] SE ESTIVER OK OU [ESC] PARA CANCELAR!"

TK = INKEY(0)

IF TK <> 13
DO QUADRO
SET COLOR TO &COR6
DO AVISO WITH 13,"IMPRESSAO CANCELADA !"
SET COLOR TO &COR1
INKEY(1)
RETURN
ENDIF

```

IF .NOT. ISPRINTER()
  DO QUADRO
  SET COLOR TO &COR6
  DO AVISO WITH 13, "ATENCAO: A IMPRESSORA NAO ESTA ATIVA "
  SET COLOR TO &COR1
  TONE(261,7,2)
  TONE(392,7)
  INKEY(1)
  RETURN
ENDIF

DO QUADRO
DO AVISO WITH 12, " A G U A R D E .... "
DO AVISO WITH 14, " TECLE <ESC> PARA INTERROMPER A IMPRESSAO "
SET COLOR TO &COR1

PG = 0
TOT= 0
FINAL = .F.
SET DEVICE TO PRINT

DO CASE

CASE OP=1
  SELE 5
  DP_CAB(TITI)

  DO WHILE .NOT. EOF()

    DO ESCPRINT

    IF FINAL
      RETURN
    ENDIF

    @ PROW()+1, 4 SAY NO
    @ PROW() , 17 SAY X
    @ PROW() , 32 SAY Y
    @ PROW() , 47 SAY Z

    IF PROW()+1 > 62
      DP_CAB(TITI)
    ENDIF

    SKIP

  ENDDO

CASE OP=2
  SELE 7
  DP_CAB(TITI)

  DO WHILE .NOT. EOF()

    DO ESCPRINT

    IF FINAL
      RETURN
    ENDIF

    @ PROW()+1, 4 SAY BAR
    @ PROW() , 17 SAY NI
    @ PROW() , 32 SAY NF
    @ PROW() , 47 SAY SECAO
    @ PROW() , 62 SAY TIPO

    IF PROW()+1 > 62
      DP_CAB(TITI)
    ENDIF

    SKIP

  ENDDO

CASE OP=3
  SELE 14
  GO TOP
  DP_CAB(TITI)

  DO WHILE .NOT. EOF()

    DO ESCPRINT

    IF FINAL
      RETURN
    ENDIF

    @ PROW()+1, 5 SAY SECAO
    @ PROW() , 15 SAY DESCRICAO

    IF PROW()+1 > 63
      DP_CAB(TITI)
    ENDIF

```

```

        SKIP
    ENDDO

CASE OP=4

SECAO1=SECAO
FIGURA1=FIGURA
DP_CAB(TIT1)
@ PROW()+2,0 SAY IA5+"FIGURA: "+FIGURA1
@ PROW()+1,0 SAY ' '

DO WHILE ((EST=ESTRU .AND. SECAO = SECAO1) .OR. VAL(SECAO2)=0) .AND. .NOT. EOF()
DO ESCPRINT

IF FINAL
RETURN
ENDIF

IF SECAO1 <> SECAO
SECAO1 = SECAO
FIGURA1 = SPAC(2)
DP_CAB(TIT1)
LOOP
ENDIF

IF FIGURA1 <> FIGURA
FIGURA1 = FIGURA
IF PROW()>57
DP_CAB(TIT1)
LOOP
ENDIF
@ PROW()+2,0 SAY IA5+"FIGURA: "+FIGURA1
@ PROW()+1,0 SAY ' '
ENDIF

DP_CAB141()

DO WHILE .NOT. EOF() .AND. EST=ESTRU .AND. SECAO=SECAO1 .AND. FIGURA=FIGURA1
DO ESCPRINT

IF FINAL
RETURN
ENDIF

@ PROW()+1,6 SAY VERT
@ PROW() ,16 SAY X
@ PROW() ,31 SAY Y
@ PROW() ,49 SAY STRZERO(NB,3)
@ PROW() ,62 SAY ESCALA

SKIP
IF PROW()+1 > 63 .AND. FIGURA=FIGURA1 .AND. .NOT. EOF()
DP_CAB(TIT1)
DP_CAB141()
ENDIF

ENDDO
REC1 = RECNO()
SKIP -1

@ PROW()+1,0 SAY ''

IF PROW()+1 > 60
DP_CAB(TIT1)
ENDIF

SELE 3
SEEK ESTRU+SECAO1+FIGURA1

DP_CAB142()

DO WHILE .NOT. EOF() .AND. EST=ESTRU .AND. SECAO=SECAO1 .AND. FIGURA=FIGURA1
DO ESCPRINT

IF FINAL
RETURN
ENDIF

@ PROW()+1,6 SAY BAR
@ PROW() ,16 SAY X
@ PROW() ,31 SAY Y
SKIP

IF PROW()+1 > 63 .AND. FIGURA=FIGURA1 .AND. .NOT. EOF()
DP_CAB(TIT1)
DP_CAB142()
ENDIF

ENDDO

```

```

        @ PROW()+1,0 SAY ''
        SELE 1
        GO RECI

    ENDDO

CASE OP=5

    SELE 6
    DP_CAB(TIT1)

    DO WHILE .NOT. EOF()
        DO ESCPRINT

            IF FINAL
                RETURN
            ENDIF
            @ PROW()+1, 4 SAY NO
            @ PROW() , 17 SAY DR
            @ PROW() , 32 SAY DESLO
            IF PROW()+1 > 62
                DP_CAB(TIT1)
            ENDIF
            SKIP

        ENDDO

    ENDCASE
EJECT
SET DEVICE TO SCREEN
RETURN

```

* DP_REL2.PRG - IMPRESSAO DOS CARREGAMENTOS DOS PILARES

```

REL2- 1

DECLARE TP2[4]

TP2[1] = " COMB.-RES. "
TP2[2] = " COMBINACAO "
TP2[3] = " ACOES-RES. "
TP2[4] = " ACOES "

DO WHILE .T.

    X1=10
    X2=13
    Y1=49
    Y2=60

    DO MENU WITH 'TP2','REL2'

    TELA1 = SAVESCREEN(7,1,21,78,)

    ESTRU2 = ESTRU

    IF OP>0
        X1=10
        X2=16
        Y1=12
        Y2=65

        DO SUJA
        DO QUADRO
    ENDIF

    DO CASE

        CASE OP = 0
            RETURN
        OTHER
            DO DP_REL21
    ENDCASE

    RESTSCREEN(7,1,21,78,TELA1)

ENDDO

```

* DP_REL21 - IMPRESSÃO DOS CARREGAMENTOS DOS PILARES

```

GO TOP
SET CONSOLE OFF

```

```

X1=10
X2=16
Y1=08
Y2=69

DO SUJA

DO CASE

CASE OP=1
  TITI='RELATORIO RESUMIDO DAS COMBINACOES'
CASE OP=2
  TITI='RELATORIO DAS COMBINACOES'
  DO QUADRO

  DO AVISO WITH 10,TITI+' DO PILAR '+ESTRU

  DO WHILE .T.

    DO AVISO WITH 16,'TECLE: [ENTER] LISTAR TUDO [ESC] CANCELAR'
    COMB2=SPAC(2)
    SET CURSOR ON
    @ 12,09 SAY 'COMBINACAO:' GET COMB2 PICT '99'
    READ
    SET CURSOR OFF

    IF LASTKEY()=27
      RESTSCREEN(7,1,21,78,TELA1)
      RETURN
    ENDIF

    COMB2=STRZERO(VAL(COMB2),2)

    IF VAL(COMB2) > 0

      SELE 13
      SEEK ESTRU+COMB2

      IF EOF()

        TONE(261,7,2)
        TONE(392,7)
        DO AVISO WITH 16,SPAC(60)
        SET COLOR TO &COR6
        DO AVISO WITH 16,'COMBINACAO NAO CADASTRADA !'
        INKEY(2)
        SET COLOR TO &COR5
        LOOP

      ELSE

        @ 14,09 SAY 'DESCRICAO: '+DP_COMB->DESCRICAO
        DO AVISO WITH 16,SPAC(60)
        DO AVISO WITH 16,'TECLE: [ENTER] PARA PROSSEGUIR [ESC] ALTERAR'
        K=INKEY(0)

        IF LASTKEY()=27
          LOOP
        ENDIF

      ENDIF

    ELSE

      SELE 13
      GO TOP

    ENDIF

  EXIT

ENDDO

CASE OP=3
  TITI='RELATORIO RESUMIDO DOS CARREGAMENTOS'

CASE OP=4
  TITI='RELATORIO DOS CARREGAMENTOS'

  DO QUADRO

  DO AVISO WITH 10,TITI+' DO PILAR '+ESTRU

  DO WHILE .T.

    DO AVISO WITH 16,'TECLE: [ENTER] LISTAR TUDO [ESC] CANCELAR'
    CARGA2=SPAC(2)
    SET CURSOR ON
    @ 12,09 SAY 'CARREGAMENTO:' GET CARGA2 PICT '999'
    READ
    SET CURSOR OFF

    IF LASTKEY()=27
      RESTSCREEN(7,1,21,78,TELA1)
      RETURN
    ENDIF
  
```

```

CARGA2=STRZERO(VAL(CARGA2),2)
IF VAL(CARGA2) > 0
  SELE 11
  SEEK ESTRU+CARGA2

  IF EOF()

    TONE(261,7,2)
    TONE(392,7)
    DO AVISO WITH 16,SPAC(60)
    SET COLOR TO &COR6
    DO AVISO WITH 16,"CARGA NAO CADASTRADA !"
    INKEY(2)
    SET COLOR TO &COR5
    LOOP

  ELSE

    @ 14,09 SAY 'DESCRICAO: '+DP_CARGA->DESCRICAO
    DO AVISO WITH 16,SPAC(60)
    DO AVISO WITH 16,"TECLE: [ENTER] PARA PROSEGUIR [ESC] ALTERAR"
    K=INKEY(0)

    IF LASTKEY()=27
      LOOP
    ENDIF

  ENDIF

ELSE

  SELE 11
  GO TOP

ENDIF

EXIT

ENDDO

ENDCASE

DO QUADRO

DO AVISO WITH 11,TIT1+" DO PILAR "+ESTRU
DO AVISO WITH 13,"VERIFIQUE A IMPRESSORA E TECLE:"
DO AVISO WITH 15,"[ENTER] SE ESTIVER OK OU [ESC] PARA CANCELAR!"

TK = INKEY(0)

IF TK <> 13
  DO QUADRO
  SET COLOR TO &COR6
  DO AVISO WITH 13,"IMPRESSAO CANCELADA !"
  SET COLOR TO &COR1
  INKEY(1)
  RETURN
ENDIF

IF .NOT. ISPRINTER()
  DO QUADRO
  SET COLOR TO &COR6
  DO AVISO WITH 13,"ATENCAO: A IMPRESSORA NAO ESTA ATIVA ."
  SET COLOR TO &COR1
  TONE(261,7,2)
  TONE(392,7)
  INKEY(1)
  RETURN
ENDIF

DO QUADRO
DO AVISO WITH 12, " A G U A R D E .... "
DO AVISO WITH 14, " TECLE <ESC> PARA INTERROMPER A IMPRESSAO "
SET COLOR TO &COR1

PG = 0
TOT= 0
FINAL = .F.

SET DEVICE TO PRINT

DO CASE

CASE OP=1
  SELE 12
  GO TOP
  DP_CAB(TIT1)

  DO WHILE .NOT. EOF()
    DO ESCPRINT

    IF FINAL
      RETURN
    ENDIF

```

```

        @ PROW()+1, 6 SAY COMB
        @ PROW() , 15 SAY DESCRICAO

        IF PROW()+1 > 63
            DP_CAB(TITI)
        ENDIF

        SKIP

    ENDDO

CASE OP=2

    COMBI=COMB
    DP_CAB(TITI)
    @ PROW()+2,0 SAY IA5+COMBI+' '+DP_COMB->DESCRICAO
    @ PROW()+1,0 SAY ' '

    DO WHILE ((EST=ESTRU .AND. COMB = COMBI) .OR. VAL(COMB2)=0) .AND. .NOT. EOF()

        DO ESCPRINT

        IF FINAL
            RETURN
        ENDIF

        IF COMBI <> COMB
            COMBI = COMB

            IF PROW()>60
                DP_CAB(TITI)
                LOOP
            ENDIF

            @ PROW()+2,0 SAY IA5+COMBI+' '+DP_COMB->DESCRICAO
            @ PROW()+1,0 SAY ' '

        ENDIF

        @ PROW()+1,6 SAY CARGA
        @ PROW() ,16 SAY DP_CARGA->DESCRICAO
        @ PROW() ,53 SAY GAMA_F
        SKIP

        IF PROW()+1 > 63 .AND. CARGA=CARGA1 .AND. .NOT. EOF()
            DP_CAB(TITI)
        ENDIF

    ENDDO

CASE OP=3

    SELE 11
    DP_CAB(TITI)

    DO WHILE .NOT. EOF()

        DO ESCPRINT

        IF FINAL
            RETURN
        ENDIF

        @ PROW()+1, 4 SAY CARGA
        @ PROW() , 17 SAY DESCRICAO

        IF PROW()+1 > 63
            DP_CAB(TITI)
        ENDIF

        SKIP

    ENDDO

CASE OP=4

    CARGA1=CARGA
    DP_CAB(TITI)
    DP_CAB241()

    DO WHILE ((EST=ESTRU .AND. CARGA = CARGA1) .OR. VAL(CARGA2)=0) .AND. .NOT. EOF()
        DO ESCPRINT

        IF FINAL
            RETURN
        ENDIF

        IF CARGA1 <> CARGA
            CARGA1 = CARGA
            DP_CAB(TITI)
            DP_CAB241()
            LOOP
        ENDIF
    
```



```

SELE 8
SEEK ESTRU+CARGA1

DO WHILE .NOT. EOF() .AND. EST=ESTRU .AND. CARGA=CARGA1

    DO ESCPRINT

        IF FINAL
            RETURN
        ENDIF

        @ PROW()+1,4 SAY NO
        @ PROW() ,17 SAY DR
        @ PROW() ,31 SAY INTEN

        SKIP
        IF PROW()+1 > 63 .AND. CARGA=CARGA1 .AND. .NOT. EOF()
            DP_CAB(TITI)
            DP_CAB241()
        ENDIF

    ENDDO

    @ PROW()+1,0 SAY ''

    IF PROW()+1 > 60
        DP_CAB(TITI)
    ENDIF

    DP_CAB242()

SELE 9
SEEK ESTRU+CARGA1

DO WHILE .NOT. EOF() .AND. EST=ESTRU .AND. CARGA=CARGA1

    DO ESCPRINT

        IF FINAL
            RETURN
        ENDIF

        @ PROW()+1,6 SAY BAR
        @ PROW() ,16 SAY DR
        @ PROW() ,31 SAY IN1
        @ PROW() ,46 SAY FIM
        @ PROW() ,61 SAY INTEN
        SKIP
        IF PROW()+1 > 63 .AND. CARGA=CARGA1 .AND. .NOT. EOF() .AND. EST=ESTRU
            DP_CAB(TITI)
            DP_CAB242()
        ENDIF

    ENDDO

SELE 11
SKIP
ENDDO

ENDCASE
EJECT

SET DEVICE TO SCREEN

RETURN

```

* DP_REL3.PRG - IMP. DOS RESULTADOS DO DIMENSIONAMENTO

```

REL3- 1

DECLARE TP2[3]

TP2[1] = " DESLOCAMENTOS "
TP2[2] = " ESFORCOS      "
TP2[3] = " ARMADURAS      "

DO WHILE .T.

    X1=10
    X2=12
    Y1=49
    Y2=63

    DO MENU WITH 'TP2','REL3'

    TELA1 = SAVESCREEN(7,1,21,78,)

    ESTRU2 = ESTRU

    IF OP>0

        SELE 10

```

```

        IF EST#ESTRU
            SEEK ESTRU
        ENDIF

        X1=10
        X2=16
        Y1=08
        Y2=69

        DO SUJA
        DO QUADRO
    ENDIF

    DO CASE

        CASE OP = 0
            RETURN
        OTHER
            DO DP_REL31

    ENDCASE

    RESTSCREEN(7,1,21,78,TELA1)
ENDDO

```

* DP_REL31 - IMPRESSÃO DOS RESULTADOS DO DIMENS. DO PILAR

```

SET CONSOLE OFF

X1=10
X2=16
Y1=08
Y2=69

DO SUJA
DO CASE

    CASE OP=1
        TITI='RELATORIO DOS DESLOCAMENTOS'
        VARI='DESLO'

    CASE OP=2
        TITI='RELATORIO DOS ESFORCOS'
        VARI='ESFORCOS'

    CASE OP=3
        TITI='RELATORIO DAS ARMADURAS'
        VARI='ARMADURAS'

    ENDCASE

    DO QUADRO

    DO AVISO WITH 11,TITI+" DO PILAR "+ESTRU
    DO AVISO WITH 13,"VERIFIQUE A IMPRESSORA E TECLE:"
    DO AVISO WITH 15,"[ENTER] SE ESTIVER OK OU [ESC] PARA CANCELAR!"

    TK = INKEY(0)

    IF TK <> 13
        DO QUADRO
        SET COLOR TO &COR6
        DO AVISO WITH 13,"IMPRESSAO CANCELADA !"
        SET COLOR TO &COR1
        INKEY(1)
        RETURN
    ENDIF

    IF .NOT. ISPRINTER()
        DO QUADRO
        SET COLOR TO &COR6
        DO AVISO WITH 13,"ATENCAO: A IMPRESSORA NAO ESTA ATIVA "
        SET COLOR TO &COR1
        TONE(261,7,2)
        TONE(392,7)
        INKEY(1)
        RETURN
    ENDIF

    DO QUADRO
    DO AVISO WITH 12," A G U A R D E .... "
    DO AVISO WITH 14," TECLE <ESC> PARA INTERROMPER A IMPRESSAO "
    SET COLOR TO &COR1

    PG = 0
    TOT= 0
    FINAL = .F.

    SET DEVICE TO PRINT

```

```

LINHAS=MLCOUNT(&VARI,78,0,.F.)
DP_CAB(TIT1)
I=1
DO WHILE I < LINHAS+1
    DO ESCPRINT
    IF FINAL
        RETURN
    ENDIF

    @ PROW()+1, 0 SAY MEMOLINE(&VARI,78,1,0,.F.)
    I=I+1

    IF PROW()+1 > 63
        DP_CAB(TIT1)
    ENDIF

ENDDO

EJECT

SET DEVICE TO SCREEN

RETURN

```

***DP_ROT.PRG - MÓDULO DE ROTINAS DO SISTEMA DP**

```

PROCEDURE SUJA
* OBJETIVO... PREENCHE COM O CHR176 (°) A AREA DE DADOS DA TELA

SET COLOR TO &COR1
@ 07,01,21,78 BOX *****
RETURN

```

```

PROCEDURE AVISO
* OBJETIVO...CENTRALIZAR UM AVISO EM UMA LINHA DETERMINADA

```

```

PARAMETERS LINHA,AVISO1
PRIVATE VTAB
VTAB = (80-LEN(AVISO1))/2

IF LINHA=23
    @ LINHA,01 SAY SPAC(78)
ENDIF

@ LINHA,VTAB SAY AVISO1

RETURN

```

```

PROCEDURE ESCPRINT
* OBJETIVO...INTERROMPER A IMPRESSAO ATRAVES DA TECLA <ESC>

```

```

PRIVATE ESC
STORE .F. TO FINAL
ESC = 1

```

```

DO WHILE ESC<>0 .AND. ESC<>27
    ESC = INKEY()
ENDDO

```

```

IF ESC = 27

```

```

    TONE(261,7,2)
    TONE(392,7)
    SET DEVICE TO SCREEN
    CONFIRME = SPACE(1)
    SET CURSOR ON
    SET COLOR TO &COR5
    @ 16,23 SAY "INTERROMPER A IMPRESSAO [S/N] " GET CONFIRME PICTURE "!" VALID(CONFIRMES*SN)
    READ
    SET CURSOR OFF

```

```

    IF CONFIRME = "N"
        @ 16,8 CLEAR TO 16,69
        SET COLOR TO &COR1
        SET DEVICE TO PRINT
        RETURN
    ENDIF

```

```

SET DEVICE TO PRINT
FINAL = .T.
EJECT

```

```

SET DEVICE TO SCREEN
DO QUADRO
SET COLOR TO &COR6
DO AVISO WITH 13,"IMPRESSAO INTERROMPIDA"
SET COLOR TO &COR1
INKEY(2)
RETURN

```

```
ENDIF
```

```
RETURN
```

```

PROC MENU
* CRIA MENUS COM SOMBRA

```

```

PARA VET,VAR
SET COLOR TO &COR3
SI=SPAC(Y2+3-Y1)
SET COLOR TO &COR4
@ X2+2,Y1 SAY SI
@ X1,Y2+2 CLEAR TO X2+1,Y2+2
SET COLOR TO &COR5
@ X1-1,Y1-1 TO X2+1,Y2+1
OP = ACHOICE(X1,Y1,X2,Y2,&VET,,T,'FUNCAO0',&VAR)
&VAR=OP
SET COLOR TO &COR1
RETURN

```

```
FUNCTION FUNCAO- 0
```

```
PARAMETERS MODO,PONTEIRO
```

```

IF LASTKEY() = 27
RETURN(0)
ENDIF

```

```

IF MODO = 1
KEYBOARD CHR(3)
RETURN(2)
ENDIF

```

```

IF MODO = 2
KEYBOARD CHR(18)
RETURN(2)
ENDIF

```

```

IF LASTKEY() = 13
RETURN(1)
ENDIF

```

```

IF LASTKEY() = 4
DP1= IIF(DP1=4,1,DP1+1)
KEYBOARD CHR(27)+CHR(13)
ENDIF

```

```

IF LASTKEY() = 19
DP1= IIF(DP1=1,4,DP1-1)
KEYBOARD CHR(27)+CHR(13)
ENDIF

```

```
RETURN(2)
```

```

PROC QUADRO
* MONTA QUADRO DE DIÁLOGO NA TELA

```

```

SET COLOR TO &COR3
SI=SPAC(Y2+3-Y1)
SET COLOR TO &COR4
@ X2+2,Y1 SAY SI
@ X1,Y2+2 CLEAR TO X2+1,Y2+2
SET COLOR TO &COR5
@ X1-1,Y1-1 CLEAR TO X2+1,Y2+1
@ X1-1,Y1-1 TO X2+1,Y2+1

```

```
RETURN
```

```

FUNCTION DP_CAB(TITULO)
* IMPRIME O CABECALHO DE RELATÓRIOS

```

```

ESTRU3 = SPACE((40 - LEN(ALLTRIM(ESTRU+' ' +DP_EST->DESCRICA0)))/2)+ALLTRIM(ESTRU+' ' +
+DP_EST->DESCRICA0)

```

```
TIT = SPACE((40 - LEN(ALLTRIM(TITULO)))/2)+TITULO
```

```

SET DEVICE TO PRINT
SET CONSOLE OFF
PG = PG + 1

```

```

@ 01,00 SAY 'EMISSAO...:'
@ 01,12 SAY DATE()
@ 01,65 SAY 'PAGINA NO.' + STRZERO(PG,3)
@ 02,00 SAY REPLICATE("-",80)
@ 03,01 SAY IA5 + ESTRU3
@ 04,01 SAY IA5 + TIT
@ 05,00 SAY REPLICATE("-",80)

```

```
IF PROCNAME(1)='DP_REL11.PRG'
```

```
DO CASE
```

```
  CASE OP=1
    DP_CAB11()
```

```
  CASE OP=2
    DP_CAB12()
```

```
  CASE OP=3
    DP_CAB13()
```

```
  CASE OP=4
    DP_CAB14()
```

```
  CASE OP=5
    DP_CAB15()
```

```
ENDCASE
```

```
ELSEIF PROCNAME(1)='DP_REL12.PRG'
```

```
DO CASE
```

```
  CASE OP=1
    DP_CAB21()
```

```
  CASE OP=2
    DP_CAB22()
```

```
  CASE OP=3
    DP_CAB23()
```

```
  CASE OP=4
    DP_CAB24()
```

```
ENDCASE
```

```
ENDIF
```

```
@ PROW()+1,00 SAY REPLICATE("-",80)
```

```
RETURN(T.)
```

```
FUNCTION DP_CAB11()
```

```
* IMPRIME O CABECALHO DE RELATÓRIOS DOS NOS
```

```
@ 06,04 SAY "NO"
@ 06,15 SAY "COORD. X (M)"
@ 06,30 SAY "COORD. Y (M)"
@ 06,45 SAY "COORD. Z (M)"
```

```
RETURN(T.)
```

```
FUNCTION DP_CAB12()
```

```
* IMPRIME O CABECALHO DE RELATÓRIOS DOS NOS
```

```
@ 06,04 SAY "BARRA"
@ 06,15 SAY "NO INICIAL"
@ 06,30 SAY "NO FINAL"
@ 06,45 SAY "SECAO"
@ 06,60 SAY "TIPO"
```

```
RETURN(T.)
```

```
FUNCTION DP_CAB13()
```

```
* IMPRIME O CABECALHO DE RELATÓRIOS RESUMIDOS DAS SECOES
```

```
@ 06,04 SAY "SECOES"
@ 06,15 SAY "DESCRICAO"
```

```
RETURN(T.)
```

```

FUNCTION DP_CAB14()
* IMPRIME O CABECALHO DE RELATÓRIOS DAS SECOES

@ PROW()+1,0 SAY IA5+SECAO+' - '+DP_SECAO->DESCRICAO
SELE 4

@ PROW()+1,14 SAY IA20+'AREA = '+ALLTRIM(STR(AREA))+' CM2+' JX = '+ALLTRIM(STR(JX))+' CM4+' JY = ;
  '+ALLTRIM(STR(JY))+' CM4+' XCG = '+ALLTRIM(STR(XCG))+' CM+' YCG = '+ALLTRIM(STR(YCG))+' CM'+IA18

SELE 1
RETURN(T.)

```

```

FUNCTION DP_CAB141()
* IMPRIME O CABECALHO DE RELATÓRIOS DOS VERTICES DAS SECOES

@ PROW()+1,4 SAY REPLICATE("-",76)
@ PROW()+1,4 SAY 'VERT/FACE'
@ PROW() ,15 SAY 'COORD. X(CM)'
@ PROW() ,30 SAY 'COORD. Y(CM)'
@ PROW() ,45 SAY 'NO. BARRAS'
@ PROW() ,60 SAY 'DIM. FACE (CM)'
@ PROW()+1,4 SAY REPLICATE("-",76)

RETURN(T.)

```

```

FUNCTION DP_CAB142()
* IMPRIME O CABECALHO DE RELATÓRIOS DAS BARRAS DAS SECOES

SELE 3
@ PROW()+1,4 SAY REPLICATE("-",76)
@ PROW()+1,5 SAY 'BARRA'
@ PROW() ,15 SAY 'COORD. X (CM)'
@ PROW() ,30 SAY 'COORD. Y (CM)'
@ PROW()+1,4 SAY REPLICATE("-",76)

RETURN(T.)

```

```

FUNCTION DP_CAB15()
* IMPRIME O CABECALHO DE RELATÓRIOS DOS APOIOS

@ 06,04 SAY "APOIOS"
@ 06,15 SAY "DESCRICAO"

RETURN(T.)

```

```

FUNCTION DP_CAB21()
* IMPRIME O CABECALHO DE RELATÓRIOS RESUMIDOS DAS COMBINACOES DE CARGAS

@ 06,04 SAY "COMBINACAO"
@ 06,15 SAY "DESCRICAO"

RETURN(T.)

```

```

FUNCTION DP_CAB22()
* IMPRIME O CABECALHO DE RELATÓRIOS DAS COMBINACOES DE CARGAS

@ 06,04 SAY "CARGAS"
@ 06,15 SAY "DESCRICAO"
@ 06,55 SAY "COEFICIENTE"

RETURN(T.)

```

```

FUNCTION DP_CAB23()
* IMPRIME O CABECALHO DE RELATÓRIOS RESUMIDOS DAS CARGAS

@ 06,04 SAY "CARGAS"
@ 06,15 SAY "DESCRICAO"

RETURN(T.)

```

```

FUNCTION DP_CAB24()
* IMPRIME O CABECALHO DE RELATÓRIOS DAS CARGAS

@ PROW()+1,0 SAY IA5+CARGA+' - '+DESCRICAO
RETURN(T.)

```

FUNCTION DP_CAB241()

* IMPRIME O CABECALHO DE RELATÓRIOS DAS CARGAS APLICADAS AOS NOS

```
@ PROW()+2,0 SAY IA5+"CARGAS APLICADAS AOS NOS"
@ PROW()+1,0 SAY ' '
@ PROW()+1,4 SAY REPLICATE("-",76)
@ PROW()+1,04 SAY "NO"
@ PROW(),15 SAY "DIRECAO"
@ PROW(),30 SAY "INTENSIDADE (KN) (KN.M)"
@ PROW()+1,4 SAY REPLICATE("-",76)
```

RETURN(T.)

FUNCTION DP_CAB242()

* IMPRIME O CABECALHO DE RELATÓRIOS DAS CARGAS APLICADAS AS BARRAS

```
@ PROW()+2,0 SAY IA5+"CARGAS APLICADAS AS BARRAS"
@ PROW()+1,0 SAY ' '
@ PROW()+1,4 SAY REPLICATE("-",76)
@ PROW()+1,04 SAY "BARRA"
@ PROW(),15 SAY "DIRECAO"
@ PROW(),30 SAY "INICIO"
@ PROW(),45 SAY "FIM"
@ PROW(),60 SAY "INTENSIDADE (KN/M)"
@ PROW()+1,4 SAY REPLICATE("-",76)
```

RETURN(T.)

PROC TRIG

* DETERMINA O VALOR FUNÇÕES TRIGONOMÉTRICAS

PARA A,SIN,COS,TAN
J = A/90

DO CASE

```
  CASE J>=3
    JS = -1
    JC = 1
```

```
  CASE J>=2
    JS = -1
    JC = -1
```

```
  CASE J>=1
    JS = 1
    JC = -1
```

```
  OTHER
    JS = 1
    JC = 1
```

ENDCASE

IF A > 90

```
  IF A>270
    A = 360-A
  ELSE
```

```
    IF A>180
      A = A-180
    ELSE
      A = 180-A
    ENDIF
```

ENDIF

ENDIF

A = A*3.141592653589793238462643/180

SIN = JS*(A - (A**3)/6 + (A**5)/120 - (A**7)/5040)
COS = JC*(1 - (A**2)/2 + (A**4)/24 - (A**6)/720)

RETURN

PROC ROTBAR1

```
L = SQRT((X2-X1)^2+(Y2-Y1)^2)
SEN = (Y2-Y1)/L
COSS = (X2-X1)/L
YR = (Y3-Y1)*COSS-(X3-X1)*SEN
T = IIF(YR<0,-1,1)
T = IIF(VAL(FIGURA)>1,-T,T)
```

```

A1 = Y2-Y1
B1 = X1-X2
C1 = X2*Y1-X1*Y2
A2 = Y3-Y2
B2 = X2-X3
C2 = X3*Y2-X2*Y3
R1 = D*SQRT(A1^2+B1^2)-C1
R2 = -R1-2*C1
R3 = D*SQRT(A2^2+B2^2)-C2
R4 = -R3-2*C2

DP = A1*B2-A2*B1
DX1= B2*R1 - B1*R3
DX2= B2*R1 - B1*R4
DX3= B2*R2 - B1*R3
DX4= B2*R2 - B1*R4
DY1= -A2*R1 + A1*R3
DY2= -A2*R1 + A1*R4
DY3= -A2*R2 + A1*R3
DY4= -A2*R2 + A1*R4

LL = 100000000
J1=1

DO WHILE J1<=4

  J11=STR(J1,1)
  XA = DX&J11
  XA = XA/DP
  YA = DY&J11
  YA = YA/DP
  LB = T*(SQRT((XA-X1)^2+(YA-Y1)^2)+SQRT((XA-X3)^2+(YA-Y3)^2))

  IF LB <= LL
    LL = LB
    XB=XA
    YB=YA
  ENDIF

  J1 = J1+1

ENDDO
RETURN

PROC ROTBAR2

SELE 2
GIRO1 = MOD(GIRO,360)
XCG1=XCG
YCG1=YCG

SELE 1
NB1 =NB
ESCALA1=ESCALA

IF GIRO1 < 0
  GIRO1 = 360+GIRO1
ENDIF

ANG = 360/NB1

DO TRIG WITH ANG/2,SIN,COS,TAN

D1=-D

IF FIGURA1='01'
  D1=-D
ENDIF

RAIO = (ESCALA1/2+D1)
W = IIF(MOD(NB1+2,4)=0,(NB1+2)/4,INT((NB1+2)/4)+1)
G = 180*(2*W-1)/NB1-90
I = 0

SELE 3

SET FILTER TO SECAO= SECAO1 .AND. FIGURA = FIGURA1 .AND. EST=ESTRU
DELE ALL

DO WHILE I <= NB1-I
  ANG1=MOD(ANG*I+G+GIRO1,360)
  DO TRIG WITH ANG1,SIN,COS,TAN
  APPEND BLANK
  X1 = ROUND(XCG1+COS*RAIO,8)
  Y1 = ROUND(YCG1+SIN*RAIO,8)

  REPLA EST WITH ESTRU,SECAO WITH SECAO1,FIGURA WITH FIGURA1,BAR WITH STRZERO(I+1,3), ;
  X WITH X1,Y WITH Y1

  I = I + 1
ENDDO
RETURN

```



```

PROC ROTBAR3

J2 = 1

DO WHILE J2 <= NBF
  I = I + 1
  XA = XBA + J2 * (XBB - XBA) / (NBF + 1)
  YA = YBA + J2 * (YBB - YBA) / (NBF + 1)
  APPEND BLANK
  REPLA EST WITH ESTRU, SECAO WITH SECAO1, FIGURA WITH FIGURA1, FACE WITH VERTI, BAR WITH TRZERO(1,3),;
  X WITH XA, Y WITH YA
  J2 = J2 + 1
ENDDO

RETURN

```

```

FUNCTION FUNCAO1

PARAMETERS MODO, PONTEIRO
VAR = VETOR1[PONTEIRO]

IF LASTKEY() = 27
  RETURN (0)
ENDIF

IF MODO = 1
  TONE(128,8)
ENDIF

IF MODO = 2
  TONE(128,8)
ENDIF

IF LASTKEY() = 13
  SET CURSOR ON
  @ ROW(), COL() GET &VAR
  READ
  SET CURSOR OFF
  IF LASTKEY() = 13

    IF EOF()
      KEYBOARD CHR(13)
    ELSE
      KEYBOARD CHR(24) + CHR(13)
    ENDIF

  ENDIF
ENDIF

RETURN(1)

```

```

FUNCTION FUNCAO2
*OBJETIVO...: GERENCIAR CARREGAMENTOS E SUAS COMBINAÇÕES

PARAMETERS MODO, PONTEIRO
VAR = VETOR1[PONTEIRO]
VAR1 = VETOR1[1]

IF LASTKEY() = 27
  RETURN (0)
ENDIF

IF LASTKEY() = 7
  VAR2 = &VAR1

  X1 = 10
  X2 = 16
  Y1 = 12
  Y2 = 65

  DO SUJA
  DO QUADRO

  IF DBF() = 'DP_COMB'
    VAR3 = 'COMBINAÇÃO DE CARREGAMENTOS'
  ELSEIF DBF() = 'DP_CARGA'
    VAR3 = 'CARREGAMENTO'
  ELSEIF DBF() = 'DP_EST'
    VAR3 = 'PILAR'
  ENDIF

  DO AVISO WITH X1, 'ELIMINAÇÃO DE ' + VAR3

  SET CURSOR ON
  SN = ''

  @ 12,13 SAY 'CODIGO...' GET &VAR1
  @ 14,13 SAY 'DESCRICAÇÃO' GET DESCRICAÇÃO

```

```

CLEAR GETS
@ 16,13 SAY ' PROSSEGUIR COM A ELIMINACAO? [S/N]:' GET SN PICT'@!'
READ
SET CURSOR OFF

IF SN = 'S'
  ESTRUI=ESTRU

  IF DBF()='DP_EST' .AND. EST=ESTRU
    ESTRUI=SPAC(2)
  ENDIF

  IF .NOT. EOF()
    SKIP
    REG=RECNO()
    SKIP -1
    DELE
    GO REG
  ELSE
    SKIP -1
    REG=RECNO()
    SKIP
    DELE
    GO REG
  ENDIF

  IF DBF()='DP_CARGA'
    SELE 8
    DELE ALL FOR EST=ESTRU .AND. CARGA=VAR2

    SELE 9
    DELE ALL FOR EST=ESTRU .AND. CARGA=VAR2

    SELE 11
  ENDIF

  IF DBF()='DP_COMB'
    SELE 13
    DELE ALL FOR EST=ESTRU .AND. COMB=VAR2
    SELE 12
  ENDIF

  IF DBF()='DP_EST'
    I=1
    DO WHILE I<=14
      IF I=10
        I=I+1
      ENDIF

      II=STR(I,2)
      SELE &II
      DELE ALL FOR EST=ESTRU

      IF ESTRUI=SPAC(2)
        SET FILTER TO
      ENDIF

      I=I+1
    ENDDO

    IF ESTRUI=SPAC(2)
      ESTRUI=ESTRUI
      DESC_ATI='NENHUMA'
    ENDIF
  ENDIF
ENDIF

ENDIF

KEYBOARD CHR(20)
SELE 10
RETURN(2)

ENDIF

IF LASTKEY() = 22
  VAR2= STRZERO(VAL(&VAR1)+1,2)
  SKIP
  IF VAR2#&VAR1
    APPEND BLANK
    IF DBF()='DP_EST'
      REPLA &VAR1 WITH VAR2,EST WITH ESTRU
    ELSE
      REPLA &VAR1 WITH VAR2,FCK WITH 20,TRECHOS WITH 1
    ENDIF
  ELSE
    SKIP -1
  ENDIF
  KEYBOARD CHR(6)
ENDIF
ENDIF

```

```

IF MODO = 1
  TONE(128,8)
ENDIF

IF MODO = 2

  VAR2 = &VARI
  APPEND BLANK

  IF DBF()# 'DP_EST'
    REPLA EST WITH ESTRU,&VARI WITH STRZERO(VAL(VAR2)+1,2)
  ELSE
    REPLA &VARI WITH STRZERO(VAL(VAR2)+1,2)
  ENDIF

  KEYBOARD CHR(6)

ENDIF

IF LASTKEY()=13 .AND. PONTEIRO=1 .AND. .NOT. BOF()

  TELA2=SAVESCREEN(7,1,23,78)
  @ 23,01 CLEAR TO 23,78

  IF DBF()= 'DP_CARGA'

    CARGA1=CARGA
    DO DP_EST41.PRG
    SELE 11

  ELSEIF DBF()= 'DP_COMB'

    COMB1=COMB
    SELE 13
    SET FILTER TO EST=ESTRU .AND. COMB=COMB1
    GO TOP
    PT=3
    KEYBOARD CHR(1)

    DECLARE VETOR4[3],VETOR5[3]

    VETOR5[1] = "CARGA "
    VETOR5[2] = "DESCRICAO"
    VETOR5[3] = "COEFICIENTE DE PONDERACAO "

    VETOR4[1] = "CARGA"
    VETOR4[2] = "DP_CARGA->DESCRICAO"
    VETOR4[3] = "GAMA_F"

    DBEDIT(07,1,21,78,VETOR4,"FUNCAO3",.T.,VETOR5)

    SELE 12

  ELSEIF DBF()= 'DP_EST'

    X1=10
    X2=16
    Y1=12
    Y2=65
    DO SUJA
    DO QUADRO
    SET CURSOR ON
    SN = ''

    DO AVISO WITH X1,'SELECAO DO PILAR ATIVO'
    @ 12,13 SAY ' CODIGO...:' GET &VARI
    @ 14,13 SAY ' DESCRICAO:' GET DESCRICAO
    CLEAR GETS
    @ 16,13 SAY ' DESEJA TORNAR ESTE PILAR ATIVO? [S/N]:' GET SN PICT '@!' VALID SN $ 'SN'
    READ
    SET CURSOR OFF

    IF SN = 'S' .AND. EST#ESTRU
      I=1

      DO WHILE I<=14

        IF I=10
          I=1+1
        ENDIF

        II=STR(I,2)
        SELE &II
        SET FILTER TO EST=ESTRU
        GO TOP
        I=1+1

      ENDDO

    ENDIF
    SELE 10
    ESTRU=EST
    DESC_ATI=DESCRICAO

  ENDIF
ENDIF

```

```

    RESTSCREEN(7,1,23,78,TELA2)
    KEYBOARD CHR(1)

ENDIF

IF ((LASTKEY() >=32 .AND. LASTKEY()<=126) .OR. LASTKEY()=13).AND.PONTEIRO=2

    IF LASTKEY() # 13
        KEYBOARD CHR(LASTKEY())
    ENDIF

    IF BOF()
        APPEND BLANK
    ENDIF

    IF DBF()# 'DP_EST'
        REPLA EST WITH ESTRU,&VARI WITH '01'
    ELSE
        REPLA &VARI WITH '01'
    ENDIF

ENDIF

SET CURSOR ON

@ ROW(),COL() GET &VAR PICT'@K'
READ
SET CURSOR OFF

DO CASE

    CASE LASTKEY()=13
        KEYBOARD CHR(6)

    CASE LASTKEY() = 3
        KEYBOARD CHR(3)

    CASE LASTKEY() = 18
        KEYBOARD CHR(18)

    CASE LASTKEY()=24
        KEYBOARD CHR(24)

    CASE LASTKEY()=5
        KEYBOARD CHR(5)

ENDCASE

ENDIF

IF PONTEIRO =1

    IF DBF()='DP_EST'
        DO AVISO WITH 23,"[ENTER] SELECIONAR == PILAR ATIVO: "+ALLTRIM(ESTRU)+" - "+ALLTRIM(DESC_ATI)
    ENDIF

    IF DBF()='DP_COMB'
        DO AVISO WITH 23,"TECLE: [ENTER] EDITAR COMBINACAO DE CARRAGAMENTOS"
    ENDIF

    IF DBF()='DP_CARGA'
        DO AVISO WITH 23,"TECLE: [ENTER] EDITAR CARRAGAMENTO"
    ENDIF

ENDIF

IF PONTEIRO = 2
    DO AVISO WITH 23,"TECLE: [ENTER] EDITAR [INSERT] CRIAR [DELETE] APAGAR [ESC] VOLTAR"
ENDIF

RETURN(1)

FUNCTION FUNCAO3
*OBJETIVO...: EDITAR NOS,BARRAS E CARRAGAMENTOS

PARAMETERS MODO,PONTEIRO
VAR = VETOR4[PONTEIRO]
VARI = VETOR4[1]

IF LASTKEY() = 27
    RETURN (0)
ENDIF

IF LASTKEY() = 7.AND. (ALLTRIM(DP->FIGURA)#'C' .OR. DBF()# 'DP_COORD')

    IF .NOT. EOF()

        SKIP
        REG=RECNO()
        SKIP -1
        DELE
        GO REG
    ENDIF
ENDIF

```

```

ELSE
    SKIP -1
    REG=RECNO()
    SKIP
    DELE
    GO REG

ENDIF

KEYBOARD CHR(20)

ENDIF

IF LASTKEY() = 22 .AND. (ALLTRIM(DP->FIGURA) #'C' .OR. DBF() #'DP_COORD')

    INC=1
    VAR2= &VAR1
    APPEND BLANK

    IF DBF()="DP_CN".OR. DBF()="DP_CB"

        REPLA CARGA WITH CARGA1

        IF VAL(VAR2)>0
            INC=0
        ENDIF

    ENDIF

    IF DBF()="DP_COORD"
        REPLA SECAO WITH SECAO1,FIGURA WITH FIGURA1
    ENDIF

    IF DBF()="DP_COMB1"
        REPLA COMB WITH COMB1
    ENDIF

    IF DBF() = "DP_APOIO"
        IF VAL(VAR2)>0
            INC=0
        ENDIF
    ENDIF

    VAR2= STRZERO(VAL(VAR2)+INC,LEN(VAR2))

    REPLA &VAR1 WITH VAR2,EST WITH ESTRU

    SKIP
    SKIP -1
    KEYBOARD CHR(20)
    RETURN(2)

ENDIF

IF MODO = 1
    TONE(128,8)
    SKIP
    SKIP -1
ENDIF

IF MODO = 2 .AND. (ALLTRIM(DP->FIGURA) #'C' .OR. DBF() #'DP_COORD')
    VAR2 = &VAR1
    APPEND BLANK

    INC=1

    IF DBF()="DP_CN".OR. DBF()="DP_CB"
        REPLA CARGA WITH CARGA1
        INC=0
    ENDIF

    IF DBF()="DP_COORD"
        REPLA SECAO WITH SECAO1,FIGURA WITH FIGURA1
    ENDIF

    IF DBF()="DP_COMB1"
        REPLA COMB WITH COMB1
    ENDIF

    IF DBF() = "DP_APOIO"
        INC=0
    ENDIF

    REPLA EST WITH ESTRU,&VAR1 WITH STRZERO(VAL(VAR2)+INC,LEN(VAR2))

    KEYBOARD CHR(20)
ENDIF

IF ((LASTKEY() >=32 .AND. LASTKEY()<=126) .OR. LASTKEY()=13) .AND. (PONTEIRO#2 .OR. DBF()#"DP_COMB1")
    IF LASTKEY() # 13
        KEYBOARD CHR(LASTKEY())
    ENDIF

```

```

IF BOF()

    APPEND BLANK
    REPLA EST WITH ESTRU

    IF DBF()="DP_CN".OR. DBF()="DP_CB"
        REPLA CARGA WITH CARGA1
    ENDIF

    IF DBF()="DP_COORD"
        REPLA SECAO WITH SECAO1,FIGURA WITH FIGURA1
    ENDIF

    IF DBF()="DP_COMBI"
        REPLA COMB WITH COMBI
    ENDIF

ENDIF

SET CURSOR ON
VAR2=&VAR
@ ROW(),COL() GET VAR2 PICT'@K'
READ
SET CURSOR OFF

IF PONTEIRO = 1 .OR. (DBF()="DP_BAR".AND.(PONTEIRO=2 .OR. PONTEIRO=3 .OR. PONTEIRO=4))
    IF VAL(VAR2)>0
        REPLA &VAR WITH STRZERO(VAL(VAR2),LEN(VAR2))
    ELSE
        RETURN(1)
    ENDIF
ELSE
    REPLA &VAR WITH VAR2
ENDIF

DO CASE
CASE (LASTKEY()=13 .OR. (LASTKEY() >=32 .AND. LASTKEY()<=126)) .AND. PONTEIRO # PT
    KEYBOARD CHR(4)
CASE (LASTKEY()=13 .OR. (LASTKEY() >=32 .AND. LASTKEY()<=126)) .AND. PONTEIRO = PT
    KEYBOARD CHR(24)+CHR(1)+CHR(4)
CASE LASTKEY() = 3
    KEYBOARD CHR(3)
CASE LASTKEY() = 18
    KEYBOARD CHR(18)
CASE LASTKEY()=24
    KEYBOARD CHR(24)
CASE LASTKEY()=5
    KEYBOARD CHR(5)
ENDCASE

IF PONTEIRO =1
    RETURN(2)
ENDIF

ENDIF

DO AVISO WITH 23,"TECLE: [INSERT/ENTER] CRIAR [DELETE] APAGAR [ESC] VOLTAR"
RETURN(1)

FUNCTION FUNCAO4
*OBJETIVO... LISTAR NA TELA OS PILARES/SEÇÕES CADASTRADOS
PARAMETERS MODO
IF LASTKEY() = 27
    RETURN (0)
ENDIF

IF MODO = 1
    TONE(128,8)
ENDIF

IF MODO = 2
    TONE(128,8)
ENDIF

RETURN(1)

```

```

FUNCTION FUNCAOS
PARAMETERS MODO,PONTEIRO
VAR = VETOR4[PONTEIRO]
VARI = VETOR4[1]

IF LASTKEY() = 27
  RETURN (0)
ENDIF

IF LASTKEY() = 7

  SET SOFTSEEK ON
  VAR2=&VARI
  DELE
  KEYBOARD CHR(13)
  SEEK ESTRU+VAR2
  SET SOFTSEEK OFF

ENDIF

IF LASTKEY() = 22

  VAR2= STRZERO(VAL(&VARI)+1,3)
  APPEND BLANK
  REPLA &VARI WITH VAR2,EST WITH ESTRU
  SKIP -13
  SEEK ESTRU+VAR2

ENDIF

IF MODO = 1
  TONE(128,8)
ENDIF

IF MODO = 2
  TONE(128,8)
ENDIF

IF (LASTKEY() >=32 .AND. LASTKEY()<=126) .OR. LASTKEY()=13

  IF LASTKEY() # 13
    KEYBOARD CHR(LASTKEY())
  ENDIF

  SET CURSOR ON
  @ ROW(),COL() GET &VAR PICT'@K'
  READ
  SET CURSOR OFF

  IF PONTEIRO =1
    REPLA &VAR WITH STRZERO(VAL(&VAR),3)
  ENDIF

  DO CASE
    CASE LASTKEY()=13 .AND. PONTEIRO<>PT
      KEYBOARD CHR(4)

    CASE LASTKEY()=13 .AND. PONTEIRO=PT
      II=1
      CHAVE='CHR(19)'

      DO WHILE II <PT-1
        CHAVE=CHAVE+"*"+CHR(19)*
        II=II+1
      ENDDO

      IF .NOT. EOF()
        KEYBOARD CHR(24)+&CHAVE
      ENDIF

    CASE LASTKEY()=24
      KEYBOARD CHR(24)

    CASE LASTKEY()=5
      KEYBOARD CHR(5)

  ENDCASE

ENDIF

RETURN(1)

```

*** DP_SEC.PRG - MODULO MENU DOS ARQUIVOS**

```

SEC=1
DECLARE TP1[5]
TP1[1] = " CRIAR "
TP1[2] = " ALTERAR "
TP1[3] = " APAGAR "
TP1[4] = " LISTAR "
TP1[5] = " VISUALIZAR "

```

```

DO WHILE .T.

    X1=09
    X2=13
    Y1=27
    Y2=38

    DO MENU WITH 'TPI','SEC'

    DO CASE

        CASE OP = 0
            EXIT
            * SE FOR PRESSIONADO <ESC> FINALIZA

        CASE OP = 1
            DO DP_SEC1
            * CRIACAO DE SECOES NOVAS

        CASE OP = 2
            DO DP_SEC2
            * ALTERACAO DE SECOES EXISTENTES

        CASE OP = 3
            DO DP_SEC3
            * DELECAO DE SECOES

        CASE OP = 4
            * LISTAGEM DAS SECOES NA TELA

            SELE 14
            SET FILTER TO EST=ESTRU
            GO TOP
            DECLARE VETOR1[2],VETOR3[2]

            VETOR3[1] = "SECAO"
            VETOR3[2] = "DESCRICAO"

            VETOR1[1] = "SECAO"
            VETOR1[2] = "DESCRICAO"

            SET FILTER TO
            KEYBOARD CHR(1)
            DBEDIT(07,1,21,78,VETOR1,"FUNCAO4",.T.,VETOR3)

            SET CURSOR OFF

        CASE OP = 5
            DO DP_SEC5
            * VISUALIZACAO DAS SECOES NA TELA"

    ENDCASE

    DO SUJA

ENDDO
RETURN

```

* DP_SEC1.PRG

```

SET CURSOR ON

DO WHILE .T.

    X1=11
    X2=14
    Y1=32
    Y2=74

    DO QUADRO

    SECAO1 = SPAC(4)

    @ X1,32 SAY ' SECAO.....' GET SECAO1 PICT '9999'
    READ

    IF VAL(SECAO1)=0 .OR. LASTKEY()=27
        SET CURSOR OFF
        RETURN
    ENDIF

    SECAO1=STRZERO(VAL(SECAO1),4)

    SELE 14
    SEEK ESTRU+SECAO1

    IF EOF()
        APPEND BLANK
        REPLA SECAO WITH SECAO1,EST WITH ESTRU
    ENDIF

```



```

@ X1+1,32 SAY ' DESCRICAO:' GET DESCRICAO
READ

ELSE
@ X1+1,32 SAY ' DESCRICAO:' GET DESCRICAO
CLEAR GETS

ENDIF

READ

IF LASTKEY()=27
SET CURSOR OFF
RETURN
ENDIF

TELA1=SAVESCREEN(7,1,23,78)

DO WHILE .T.

X1=11
X2=14
Y1=32
Y2=74

SET CURSOR ON
FIGURA1 = SPAC(2)

@ X1+2,33 SAY 'FIGURA...:' GET FIGURA1 PICT '99'
READ

IF VAL(FIGURA1) = 0 .OR. LASTKEY()=27
DO DP_SEC12
EXIT
ENDIF

FIGURA1= STRZERO(VAL(FIGURA1),2)

SELE 2
SEEK ESTRU+SECAO1+FIGURA1

IF .NOT. FOUND()
APPEND BLANK
REPLA EST WITH ESTRU,SECAO WITH SECAO1,FIGURA WITH FIGURA1
ENDIF

TIPO1=SPAC(4)
TIPO2=SPAC(1)
N=0

@ X1+3,33 SAY 'TIPO.....:' GET TIPO1 PICT '@!' VALID LEFT(TIPO1,1)$'PICR' .AND.(LEFT(TIPO1,1)='C' ;
.OR. LEFT(TIPO1,1)='R' .OR. VAL(RIGHT(TIPO1,3))>2)
READ

TIPO2=LEFT(TIPO1,1)
N = VAL(RIGHT(TIPO1,3))

IF LASTKEY()=27
LOOP
ELSE
DELE ALL FOR SECAO=SECAO1 .AND. FIGURA=FIGURA1 .AND. EST=ESTRU
SELE 1
DELE ALL FOR SECAO=SECAO1 .AND. FIGURA=FIGURA1 .AND. EST=ESTRU
SET FILTER TO SECAO=SECAO1 .AND. FIGURA=FIGURA1 .AND. EST=ESTRU
SELE 2
APPEND BLANK
REPLA EST WITH ESTRU,SECAO WITH SECAO1,FIGURA WITH FIGURA1,TIPO WITH TIPO1
ENDIF

NB1=0

DO CASE

CASE LEFT(TIPO,1) = 'P'
X1=12
X2=16
Y1=44
Y2=72

DO QUADRO

@ X1,45 SAY 'DIMENSAO DOS LADOS:' GET ESCALA VALID(ESCALA>0)
@ X1+1,45 SAY 'X DO CG.....:' GET XCG
@ X1+2,45 SAY 'Y DO CG.....:' GET YCG
@ X1+3,45 SAY 'GIRO DA SECAO.....' GET GIRO VALID(GIRO >=-360 .AND. GIRO <=360) PICT '999.99'
@ X1+4,45 SAY 'BARRAS POR FACE.....' GET NB1 PICT '999'
READ

XCG1 = XCG
YCG1 = YCG
GIRO1 = MOD(GIRO,360)
ESCALA1=ESCALA

```

```

IF GIROI < 0
  GIROI = 360+GIROI
ENDIF

ANG = 360/N
DO TRIG WITH ANG/2,SIN,COS,TAN

RAIO = DP->ESCALA/2/SIN
W = IIF(MOD(N+2,4)=0,(N+2)/4,INT((N+2)/4)+1)
G = 180*(2*W-1)/N-90

I = 0
SELE I

DO WHILE I <= N-1

  ANGI=MOD(ANG*I+G+GIROI,360)
  DO TRIG WITH ANGI,SIN,COS,TAN
  APPEND BLANK

  X1 = ROUND(XCGI+COS*RAIO,8)
  Y1 = ROUND(YCGI+SIN*RAIO,8)

  REPLA EST WITH ESTRU,SECAO WITH SECAOI,FIGURA WITH FIGURAI,VERT WITHSTRZERO(I+1,3),
  X WITH X1, Y WITH Y1,NB WITH NBI,ESCALA WITH ESCALAI

  I = I + 1
ENDDO

CASE LEFT(TIPO,1) = 'R'
X1=12
X2=17
Y1=44
Y2=72

DO QUADRO

LX1=0
LY1=0

@ X1,45 SAY 'DIMENSAO DA BASE:?' GET LX1 VALID(LX1>0) PICT '9999.99'
@ X1+1,45 SAY 'DIMENSAO DA ALTURA?' GET LY1 VALID(LY1>0) PICT '9999.99'
@ X1+2,45 SAY 'X DO CG.....?' GET XCG
@ X1+3,45 SAY 'Y DO CG.....?' GET YCG
@ X1+4,45 SAY 'GIRO DA SECAO.....?' GET GIROI VALID(GIRO >=-360 AND GIROI <=360)
@ X1+5,45 SAY 'BARRAS POR FACE:?' GET NBI PICT '999'
READ

XCGI = XCG
YCGI = YCG
GIROI = MOD(GIRO,360)
ESCALAI=ESCALA

IF GIROI < 0
  GIROI = 360+GIROI
ENDIF

X1=-LX1/2
Y1=-LY1/2
X2=LX1/2
Y2=-LY1/2
X3=LX1/2
Y3=LY1/2
X4=-LX1/2
Y4=LY1/2

DO TRIG WITH GIROI,SIN,COS,TAN

I=1
SELE I

DO WHILE I <= 4
  I1=STR(I,1)
  XX=XCGI+X&I1*COS+Y&I1*SIN
  YY=YCGI+Y&I1*COS-X&I1*SIN
  APPEND BLANK

  REPLA EST WITH ESTRU,SECAO WITH SECAOI,FIGURA WITH FIGURAI,VERT WITH STRZERO(I,3),
  X WITH XX,Y WITH YY,NB WITH NBI

  I = I + 1
ENDDO

GO TOP
X1=X
Y1=Y
X1=X
Y1=Y
SKIP

DO WHILE .NOT. EOF()
X2=X
Y2=Y
L=SQRT((X2-X1)**2+(Y2-Y1)**2)
SKIP-1

```

```

        REPLA ESCALA WITH L
        X1=X2
        Y1=Y2
        SKIP+2
    ENDDO

    GO BOTTOM
    L=SQRT((X1-X1)**2+(Y1-Y1)**2)
    REPLA ESCALA WITH L
CASE LEFT(TIPO,1) = 'I'

    X1=15
    X2=15
    Y1=44
    Y2=68

    DO QUADRO

    @ X1,45 SAY 'BARRAS POR FACE....' GET NBI PICT '999'
    READ

    I = I

    SELE I

    DO WHILE I <= N
        APEND BLANK
        REPLA EST WITH ESTRU,SECAO WITH SECAO1,FIGURA WITH FIGURA1,VERT WITH STRZERO(1,3),
        NB WITHNBI
        I=I+1
    ENDDO

    SET FILTER TO SECAO=SECAO1 .AND. FIGURA=FIGURA1 .AND. EST=ESTRU
    GO TOP

    SET COLOR TO &COR1
    PT=4
    DECLARE VETOR4{4},VETOR5{4}

    VETOR5{1} = "NUMERO VERTICE"
    VETOR5{2} = "X VERTICE"
    VETOR5{3} = "Y VERTICE"
    VETOR5{4} = "NUMERO BARRAS FACE"

    VETOR4{1} = "VERT"
    VETOR4{2} = "X"
    VETOR4{3} = "Y"
    VETOR4{4} = "NB"

    KEYBOARD CHR(1)
    DBEDIT(07,1,21,78,VETOR4,"FUNCAO3",.T.,VETOR5)
    SET COLOR TO &COR5
    GO TOP
    X1=X
    Y1=Y
    X1=X
    Y1=Y
    SKIP

    DO WHILE .NOT. EOF()
        X2=X
        Y2=Y
        L=SQRT((X2-X1)**2+(Y2-Y1)**2)
        SKIP-1
        REPLA ESCALA WITH L
        X1=X2
        Y1=Y2
        SKIP+2
    ENDDO
    GO BOTTOM
    L=SQRT((X1-X1)**2+(Y1-Y1)**2)
    REPLA ESCALA WITH L
CASE LEFT(TIPO,1) = 'C'

    X1=12
    X2=16
    Y1=44
    Y2=72

    DO QUADRO

    @ X1,45 SAY 'DIAMETRO.....' GET ESCALA VALID(ESCALA>0)
    @ X1+1,45 SAY 'X DO CG.....' GET XCG
    @ X1+2,45 SAY 'Y DO CG.....' GET YCG
    @ X1+3,45 SAY 'QTD. DE BARRAS....' GET NBI PICT '99'
    @ X1+4,45 SAY 'GIRO DAS BARRAS....' GET GIRO VALID(GIRO>=-360 .AND. GIRO <=360)
    READ

    N=DP_EST->CIRCULO
    XCG1 = XCG
    YCG1 = YCG
    ESCALA1=ESCALA
    GIRO1=GIRO
    ANG = 360/N

```

```

DO TRIG WITH ANG/2,SIN,COS,TAN

RAIO = DP->ESCALA/2
W = IIF(MOD(N+2,4)=0,(N+2)/4,INT((N+2)/4)+1)
G = 180*(2*W-1)/N-90

I = 0
SELE 1
DELE ALL FOR SECAO=SECAO1 .AND. FIGURA = FIGURA1

ESCALA1=ESCALA1*SIN

DO WHILE I <= N-I

    ANGI=MOD(ANG*I+G,360)
    DO TRIG WITH ANGI,SIN,COS,TAN
    APPEND BLANK

    XI = ROUND(XCG1+COS*RAIO,8)
    YI = ROUND(YCG1+SIN*RAIO,8)

    REPLA EST WITH ESTRU,SECAO WITH SECAO1,FIGURA WITH FIGURA1,VERT WITH STRZERO(1+1,3), ;
    X WITH XI,Y WITH YI,NB WITH NB1,ESCALA WITH ESCALA1

    I = I + 1

ENDDO

ENDCASE

SET FILTER TO SECAO = SECAO1 .AND. FIGURA = FIGURA1 .AND. EST=ESTRU
DO DP_SEC11
SET FILTER TO
RESTSCREEN(7,1,23,78,TELA1)

ENDDO

```

***DP_SEC11.PRG**

```

SELE 10

IF EST # ESTRU
    SEEK ESTRU
ENDIF

SELE 1

D=DIST_CG
COND = ''

IF TIPO1='C'
    DO ROTBAR2
    RETURN
ENDIF

SELE 3
DELE ALL FOR SECAO=SECAO1 .AND. FIGURA=FIGURA1 .AND. EST=ESTRU

SELE 1
GO TOP
I=1
XB=0
YB=0
XF=X
YF=Y
GO BOTTOM
XI=X
YI=Y
VERT2=VERT
GO TOP

DO WHILE T.
    X1 = XI
    Y1 = YI
    NBF = NB
    VERT1 = VERT
    X2 = X
    Y2 = Y
    XI = X
    YI = Y

    IF VERT2=VERT
        X3=XF
        Y3=YF
        COND = 'E'
    ELSE
        SKIP
        X3=X
        Y3=Y
    ENDIF
ENDIF

```

```

DO ROTBARI
SELE 3
APPEND BLANK
REPLA EST WITH ESTRU,SECAO WITH SECAOI,FIGURA WITH FIGURAI,FACE WITH VERT1,;
BAR WITH STRZERO(1,3), X WITH XB,Y WITH YB

I=[I+1+NBF
SELE 1

IF COND = 'E'
EXIT
ENDIF

ENDDO

SELE 3
SET FILTER TO SECAO=SECAOI .AND. FIGURA=FIGURAI .AND. EST=ESTRU
GO BOTTOM
FACEI=FACE
GO TOP
COND = ''
XBF = X
YBF = Y
XBA = X
YBA = Y
II=0
I=0

DO WHILE .T.
VERTI = FACE

IF FACEI=FACE
I=I+1
XBB = XBF
YBB = YBF
NBF = DP_COORD->NB
COND = 'E'

ELSE
I=I+1
NBF = DP_COORD->NB

IF NBF <> 0
SKIP
ENDIF

XBB = X
YBB = Y

ENDIF

DO ROTBAR3
SKIP
XBA=XBB
YBA=YBB

IF COND = 'E'
SET FILTER TO
SELE 1
SET FILTER TO
SELE 2
RETURN
ENDIF
ENDDO

```

* DP_SEC12.PRG

```

SELE 1
SET FILTER TO
J = 1
J1 = 1
J2 = 1
S = 0
S1 = 0
SX = 0
SY = 0
SX1=0
SY1=0
JX = 0
JY = 0
JX1=0
JY1=0
SN = 1
SEEK ESTRU+SECAOI

IF EOF()
RETURN
ENDIF

DO WHILE SECAO = SECAOI
XF=X
YF=Y
FIGURAI=FIGURA

```

```

DO WHILE FIGURA=FIGURA1.AND.SECAO=SECAO1
  X1=X
  Y1=Y
  SKIP

  IF FIGURA # FIGURA1 .OR. SECAO # SECAO1 .OR. EOF()
    X2=XF
    Y2=YF
  ELSE
    X2=X
    Y2=Y
  ENDIF

  S1 = S1+(X1*Y2-X2*Y1)/2
  SX2=(X1*Y2-X2*Y1)*(Y1+Y2)/6
  SX1 = SX1+SX2
  JX2 =((3*X2+X1)*(Y2-Y1)^3+(8*X2*Y1+4*X1*Y1)*(Y2-Y1)^2+(6*X2*Y1^2+6*X1*Y1^2)*(Y2-Y1))/12
  JX1 = JX1+JX2
  SY2=(X1*Y2-X2*Y1)*(X1+X2)/6
  SY1 = SY1+SY2
  JY2= -SN*((3*Y2+Y1)*(X2-X1)^3+(8*Y2*X1+4*Y1*X1)*(X2-X1)^2+(6*Y2*X1^2+6*Y1*X1^2)*(X2-X1))/12
  JY1 = JY1+JY2

ENDDO
SKIP -1

IF (S1<0 .AND. VAL(FIGURA)=1) .OR. (S1>0 .AND. VAL(FIGURA)>1)
  SN=-1
  SELE 2
  REPLA CIRC WITH 'I'
ELSE
  SN=1
  SELE 2
  REPLA CIRC WITH 'N'
ENDIF
SELE 1
SKIP

JY=JY+SN*JY1
JX=JX+SN*JX1
SX=SX+SN*SX1
SY=SY+SN*SY1
S =S+SN*S1

S1=0
SX1=0
SY1=0
JX1=0
JY1=0

ENDDO

XCG1 = SY/S
YCG1 = SX/S
JY = JY-S*XCG1^2
JX = JX-S*YCG1^2
JY1=JY
JX1=JX
SX1=SX
SY1=SY

SELE 4

SEEK ESTRU+SECAO1

IF EOF()
  APPEND BLANK
  REPLA SECAO WITH SECAO1,EST WITH ESTRU
ENDIF

REPLA AREA WITH S,XCG WITH XCG1,YCG WITH YCG1,SX WITH SX1,SY WITH SY1,JX WITH JX1,JY WITH JY1
RETURN

```

* DP_SEC2.PRG

```

SET CURSOR ON

DO WHILE .T.

  SELE 14
  X1=11
  X2=15
  Y1=32
  Y2=74

  DO QUADRO

  SECAO1 = SPAC(4)
  @ X1,32 SAY ' SECAO.....' GET SECAO1 PICT '9999'
  READ

```

```

IF VAL(SECAO1)=0 .OR. LASTKEY()=27
  SET CURSOR OFF
  RETURN
ENDIF

SECAO1=STRZERO(VAL(SECAO1),4)

SELE 14
SEEK ESTRU+SECAO1

IF EOF()
  LOOP
ENDIF

@ X1+2,32 SAY ' DESCRICAO:' GET DESCRICAO PICT '@K'
READ

TELA1=SAVESCREEN(7,1,23,78)

DO WHILE .T.

  X1=11
  X2=15
  Y1=32
  Y2=74
  SET CURSOR ON
  FIGURA1 = SPAC(2)
  @ X1+4,32 SAY ' FIGURA...' GET FIGURA1 PICT '99'
  READ

  IF VAL(FIGURA1) = 0 .OR. LASTKEY()=27
    DO DP_SEC12
    EXIT
  ENDIF

  FIGURA1=STRZERO(VAL(FIGURA1),2)
  SELE 2
  SEEK ESTRU+SECAO1+FIGURA1

  IF EOF()
    LOOP
  ENDIF

  TIPO1=TIPO
  SELE 1
  SET FILTER TO EST= ESTRU .AND. SECAO= SECAO1 .AND. FIGURA = FIGURA1
  GO TOP
  SET COLOR TO &COR1

  DO CASE

    CASE SUBS(TIPO1,1,2) <> 'C '
      PT=4
      DECLARE VETOR4[5],VETOR5[5]

      VETOR5[1] = "VERTICE/FACE"
      VETOR5[2] = "X VERTICE (CM)"
      VETOR5[3] = "Y VERTICE (CM)"
      VETOR5[4] = "BARRAS POR FACE"
      VETOR5[5] = "DIM. FACE"

      VETOR4[1] = "VERT"
      VETOR4[2] = "X"
      VETOR4[3] = "Y"
      VETOR4[4] = "NB"
      VETOR4[5] = "ESCALA"

      KEYBOARD CHR(1)
      DBEDIT(07,1,21,78,VETOR4,"FUNCAO3",.T.,VETOR5)

    CASE SUBS(TIPO1,1,2)= 'C '
      PT=3
      DECLARE VETOR4[4],VETOR5[4]

      VETOR5[1] = "VERTICE/FACE"
      VETOR5[2] = "X"
      VETOR5[3] = "Y"
      VETOR5[4] = "DIM. FACE"

      VETOR4[1] = "VERT"
      VETOR4[2] = "X"
      VETOR4[3] = "Y"
      VETOR4[4] = "ESCALA"

      KEYBOARD CHR(1)
      DBEDIT(07,1,21,78,VETOR4,"FUNCAO3",.T.,VETOR5)

  ENDCASE

  DO DP_SEC11
  SET FILTER TO
  SET COLOR TO &COR5
  RESTSCREEN(7,1,23,78,TELA1)
ENDDO
ENDDOD

```

* DP_SEC3.PRG

```
DO WHILE .T.
  X1=14
  X2=14
  Y1=32
  Y2=45

  DO QUADRO
  SELE 14
  SECAO1 = SPAC(4)
  SET CURSOR ON

  @ 14,32 SAY ' APAGAR:' GET SECAO1
  READ

  SET CURSOR OFF

  IF SECAO1 = SPACE(4)
    EXIT
  ENDIF

  IF VAL(SECAO1)=0 .OR. LASTKEY()=27
    SET CURSOR OFF
    RETURN
  ENDIF

  SECAO1=STRZERO(VAL(SECAO1),4)

  SEEK ESTRU+SECAO1

  IF EOF()
    LOOP
  ENDIF

  TELA1=SAVESCREEN(7,1,23,78)

  X1=10
  X2=16
  Y1=12
  Y2=65

  DO SUJA
  DO QUADRO

  DO AVISO WITH X1,'ELIMINACAO DE SECAO'

  SET CURSOR ON
  SN = ''
  @ 12,13 SAY ' CODIGO... ' +SECAO
  @ 14,13 SAY ' DESCRICAO: ' +DESCRICAO
  @ 16,13 SAY ' VOCE DESEJA ELIMINAR A SECAO? [S/N]:' GET SN
  READ

  SET CURSOR OFF

  IF SN = 'S'
    DELE
    SELE 4
    DELE ALL FOR SECAO=SECAO1 .AND. EST=ESTRU
    SELE 3
    DELE ALL FOR SECAO=SECAO1 .AND. EST=ESTRU
    SELE 2
    DELE ALL FOR SECAO=SECAO1 .AND. EST=ESTRU
    SELE 1
    DELE ALL FOR SECAO=SECAO1 .AND. EST=ESTRU
  ENDIF
  RESTSCREEN(7,1,23,78,TELA1)
ENDDO
```

* DP_SEC5.PRG - EXIBE SECAO NA TELA

```
X1=14
X2=14
Y1=32
Y2=46

DO WHILE .T.
  DO QUADRO
  SELE 14
  SECAO1 = SPAC(4)
  SET CURSOR ON
  @ 14,32 SAY ' SECAO:' GET SECAO1
  READ

  SET CURSOR OFF

  IF VAL(SECAO1) = 0 .OR. LASTKEY()=27
    EXIT
  ENDIF

  SECAO1=STRZERO(VAL(SECAO1),4)
```



```

SEEK ESTRU+SECAO1

IF EOF()
  EXIT
ENDIF

DESCRICA01=DESCRICA0

SELE 1
SET FILTER TO SECAO = SECAO1 .AND. EST=ESTRU
GO TOP
SECAO2 = SECAO1+'.SEC'
SECAO3 = SECAO1+'.BAR'
SECAO4 = SECAO1+'.GEO'

COPY FIELDS FIGURA,VERT,X,Y,ESCALA TO &SECAO2 DELI WITH BLANK

SET FILTER TO

SELE 3
SET FILTER TO SECAO = SECAO1 .AND. EST=ESTRU
GO TOP

COPY FIELDS X,Y TO &SECAO3 DELI WITH BLANK
SET FILTER TO

SELE 4
SET FILTER TO SECAO=SECAO1 .AND. EST=ESTRU

COPY FIELDS AREA,XCG,YCG,SX,SY,JX,JY TO &SECAO4 DELI WITH BLANK
SET FILTER TO

TELA1=SAVESCREEN(0,0,24,79)
RUN DP_SEC51 &SECAO1
RESTSCREEN(0,0,24,79,TELA1)

ERASE &SECAO2
ERASE &SECAO3
ERASE &SECAO4

ENDDO

```

* DP_TELA: PROGRAMA PARA TELA DO SISTEMA DP

```

SET COLOR TO &COR1
CLEAR
*CABECALHO

@ 00,00 TO 04,23
@ 00,23 TO 04,70
@ 00,70 TO 04,79
@ 00,00 SAY CHR(218)
@ 00,23 SAY CHR(194)
@ 00,70 SAY CHR(194)
@ 00,79 SAY CHR(191)
*LINHA DE MENU
@ 04,00 TO 06,79
@ 04,00 SAY CHR(195)
@ 04,23 SAY CHR(193)
@ 04,70 SAY CHR(193)
@ 04,79 SAY CHR(180)
*AREA DE TRABALHO

@ 07,01,21,78 BOX *oooooooooooo*

@ 06,00 TO 22,79
@ 06,00 SAY CHR(195)
@ 06,79 SAY CHR(180)
*AREA DE MENSAGEM

@ 22,00 TO 24,79
@ 22,00 SAY CHR(195)
@ 22,79 SAY CHR(180)

DATA = DATE()
DIA = DOW(DATA)

DO CASE

CASE DIA = 1
  @ 01,71 SAY "DOMINGO"

CASE DIA = 2
  @ 01,71 SAY "SEGUNDA"

CASE DIA = 3
  @ 01,71 SAY "TERCA"

CASE DIA = 4
  @ 01,71 SAY "QUARTA"

```

```

CASE DIA = 5
  @ 01,71 SAY "QUINTA"

CASE DIA = 6
  @ 01,71 SAY "SEXTA"

CASE DIA = 7
  @ 01,71 SAY "SABADO"

ENDCASE

@ 02,71 SAY DATA
@ 03,71 SAY TIME()
SET COLOR TO &COR2
@ 01,01 SAY " ESCOLA DE ENGENHARIA "
@ 02,01 SAY " DE SAO CARLOS -- USP "
@ 03,01 SAY " ENG. NIVALDO CAMPOS "

SET COLOR TO &COR1

DO AVISO1 WITH 1,"  DIMENSIONAMENTO  "
DO AVISO1 WITH 2,"      DE      "
DO AVISO1 WITH 3,"PILARES EM CONCRETO ARMADO"
RETURN

```

```

PROCEDURE AVISO1
* OBJETIVO...CENTRALIZAR O NOME DO PROGRAMA

PARAMETERS LINHA,AVISO

LL=47

PRIVATE VTAB
VTAB = 24+INT((LL-LEN(AVISO))/2)
@ LINHA,VTAB SAY AVISO
RETURN

```

***DP_CONF.PRG - GERENCIAMENTO DE PARÂMETROS**

```

SELE 10
CONF = 1

DECLARE TPI[5]
TPI[1] = " PILARES "
TPI[2] = " SECOES "
TPI[3] = " DIMENSIONAM. "
TPI[4] = " MATERIAIS "
TPI[5] = " SISTEMA "

DO WHILE .T.

  X1=09
  X2=13
  Y1=59
  Y2=72

  DO MENU WITH 'TPI',CONF'

  IF OP>0 .AND. OP<5

    SELE 10

    IF EST#ESTRU
      SEEK ESTRU
    ENDIF

    X1=10
    X2=16
    Y1=08
    Y2=69

    DO SUJA
    DO QUADRO
  ENDIF

  DO CASE

    CASE OP = 0
      EXIT
      * SE FOR PRESSIONADO <ESC> FINALIZA

    CASE OP = 1.AND. VAL(ESTRU)>0
      TITI="EDICAO DOS PARAMETROS"
      DO AVISO WITH 10,TITI+" DO PILAR "+ESTRU
      SET CURSOR ON

```

```

@ 14,33 SAY "TRECHOS:" GET TRECHOS PICT '999'
READ
SET CURSOR OFF

CASE OP = 2 .AND. VAL(ESTRU)>0
TIT1='EDICAO DOS PARAMETROS DAS SECOES'
DO AVISO WITH 10,TIT1+' DO PILAR '+ESTRU
SET CURSOR ON
@ 14,09 SAY "POLIGONO INSCRITO..." GET CIRCULO PICT '999'
@ 14,40 SAY "COBRIMENTO DA ARMADURA:" GET DIST_CG PICT '99.99'
READ
SET CURSOR OFF

CASE OP = 3 .AND. VAL(ESTRU)>0
TIT1='EDICAO DOS PARAMETROS DE DIMENSIONAMENTO'
DO AVISO WITH 10,TIT1+' DO PILAR '+ESTRU
SET CURSOR ON
@ 13,09 SAY "TAXA MINIMA DE ARMADURA:" GET ARM_MIN PICT '9.99' VALID ARM_MIN>0
@ 15,09 SAY "TAXA MAXIMA DE ARMADURA:" GET ARM_MAX PICT '9.99' VALID ARM_MAX>0
READ
SET CURSOR OFF

CASE OP = 4 .AND. VAL(ESTRU)>0
TIT1='EDICAO DOS PARAMETROS DOS MATERIAIS'
DO AVISO WITH 10,TIT1+' DO PILAR '+ESTRU
SET CURSOR ON
@ 13,09 SAY "TIPO DO ACO..."
SET COLOR TO N/W
@ 13,25 SAY 'CA'
@ 15,25 SAY 'C'
SET COLOR TO &COR5
@ 13,27 GET FYK PICT '99' VALID RIGHT(FYK,1)='A' OR RIGHT(FYK,1)='B'
@ 13,33 SAY "COEF. MIN. ACO DIM/DEF:" GET GAMA_S PICT '9.99' VALID GAMA_S >0
@ 13,62 SAY "/"
@ 13,64 GET GAMA_SD PICT '9.99' VALID GAMA_SD > 0
@ 15,09 SAY "TIPO CONCRETO..."
@ 15,26 GET FCK PICT '999'
@ 15,33 SAY "COEF. MIN. CON DIM/DEF:" GET GAMA_C PICT '9.99' VALID GAMA_C >0
@ 15,62 SAY "/"
@ 15,64 GET GAMA_CD PICT '9.99' VALID GAMA_CD>0

READ
SET CURSOR OFF

CASE OP = 5
DO DP_INDE

ENDCASE

DO SUJA
SET COLOR TO &COR1

ENDDO
RETURN

```

DP_SEC51.BAS: APRESENTA SEÇÕES NA TELA

```

NS = COMMAND$

DIM K(100),X(100),Y(100),V(100),XB(100),YB(100),BN(100,100),DIAM(100)
R = 1
D = 2
SCREEN R
CLS

OPEN NS+" SEC" FOR INPUT AS #1
OPEN NS+" GEO" FOR INPUT AS #3

A = 0
AA = 0
K = 0
IB = 0
K(0) = 1
I = 1

DO

IF EOF(1) THEN GOTO 130
INPUT#1 ,F,V(I),X(I),Y(I),DIAM(I)
IF A<> F THEN K(AA) = K: A=F:AA = AA+1

IF AA >0 THEN
IF K(AA)<>K(AA-1)+I THEN IB = IB+1
END IF

K = K+1
I = I+1

WEND

130 K(AA)=K
IF K(AA)=K(AA-1)+I THEN IB=IB-1

```

```

OPEN N$+" BAR" FOR INPUT AS #2
I=1

DO
  IF EOF(2) THEN 650
  INPUT#2 ,XB(I),YB(I)
  I=I+1

WEND

650 NB = I-1: I=K

  INPUT#3 ,S,XCG,YCG,SX,SY,JX,JY

200 CLOSE #1,#2,#3

NX = 1000000
NY = 1000000
MX = 0
MY = 0

DM = 124
DT = 144
DX1 = 256-DT
DY1 = 200-DT

IF K(I) = 1 THEN
  MX = DIAM(1)/2
  MY = DIAM(1)/2
  NX = -DIAM(1)/2
  NY = -DIAM(1)/2
  GOTO 450
END IF

FOR I = 1 TO K(I)

  IF X(I) > MX THEN MX = X(I)
  IF X(I) < NX THEN NX = X(I)

  IF Y(I) > MY THEN MY = Y(I)
  IF Y(I) < NY THEN NY = Y(I)

NEXT

450 LP=NB
IF K(AA)>NB THEN LP=K(AA)

FOR I = 1 TO LP

  X(I) = X(I)-NX
  Y(I) = Y(I)-NY
  XB(I) = XB(I)-NX
  YB(I) = YB(I)-NY

NEXT

MX = MX - NX
MY = MY - NY

FOR I = 1 TO LP
  Y(I) = MY-Y(I)
  YB(I) = MY-YB(I)
NEXT

DX = DM/MX
DY = DM/MY

IF DX-DY >= 0 THEN SC = DY
IF DX-DY < 0 THEN SC = DX

DX = 0
DY = 0

DX = (DT-MX*SC)/2+DX1
DY = (DT-MY*SC)/2+DY1

DX1$=STR$(DX1*R*1.25)
DY1$=STR$(DY1)
DT1$=STR$(256*R*1.25-1)

LOCATE 3,2: ? "CARACTERISTICAS"
LOCATE 5,2: ? " GEOMETRICAS "

LOCATE 3,23*R: ? "SECAO " + N$

DRAW "BM 0,0 M "+DT1$+",0 M "+DT1$+",199 M 0,199 M 0,0 BM "+DX1$+",0 M "+DX1$+",199 BM 0,"+DY1$+" M
"+DT1$+", "+DY1$

I = I
K(0)=0

```

```

FOR J = 1 TO A
  XS = STR$(INT((X(I)*SC+DX)*R*1.25))
  YS = STR$(INT((Y(I)*SC+DY)))
  DRAW "BM "+XS+", "+YS

  IF K(J)=K(J-1)+1 THEN CIRCLE (VAL(XS),VAL(YS)),INT(DIAM(I)/2*SC*1.25):GOTO 250

  FOR I = K(J-1)+1 TO K(J)
    XS=STR$(INT((X(I)*SC+DX)*R*1.25))
    YS=STR$(INT((Y(I)*SC+DY)))
    DRAW "M "+XS+", "+YS

  NEXT I

  I = K(J-1)+1
  XS=STR$(INT((X(I)*SC+DX)*R*1.25))
  YS=STR$(INT((Y(I)*SC+DY)))
  DRAW "M "+XS+", "+YS
250 I = K(J)+1

NEXT J
IF MX>MY THEN SWAP MX,MY

1000 FOR I = 1 TO NB
  XS=STR$(INT((XB(I)*SC+DX)*R*1.25))
  YS=STR$(INT((YB(I)*SC+DY)))
  IF I=1 AND R=1 THEN PAINT (VAL(XS),VAL(YS)),5,7
  DRAW "BM "+XS+", "+YS
  DRAW "NM +1,0 NM -1,0 NM +0,1 NM +0,-1"
NEXT

L=09
LOCATE L,02
? "AREA =": : ? USING "#####" ;S
LOCATE L+4,2
? "MS(X) =": : ? USING "#####" ;SX
LOCATE L+5,2
? "MS(Y) =": : ? USING "#####" ;SY
LOCATE L+9,2
? "X(CG) =": : ? USING "#####" ;XCG
LOCATE L+10,2
? "Y(CG) =": : ? USING "#####" ;YCG
LOCATE L+14,2
? "JX(CG) =": : ? USING "#####" ;JX
LOCATE L+15,2
? "JY(CG) =": : ? USING "#####" ;JY;

AS = INPUT$(1)
SCREEN 0
END

```

* PT.FOR : DETERMINA ESFORÇOS EM PILARES

```

PARAMETER(INN=10,NX=6*INN)

REAL X(INN),Y(INN),Z(INN),A1,C,P2,PB1,L1,S,CC,SB,CB,LDELTI
INTEGER NI(INN),NF(INN),NSECAO(INN),ND,NL,OP,LDELTDRT
REAL ASEC(INN),L(INN),CX(INN),CY(INN),CZ(INN),JX(INN),JY(INN)
REAL KK(NX,NX),KB(12,12),T(12,12),KP(12,12),SS(12,12)
INTEGER ID(INN,6)
REAL P(NX),PB(12),PP(NX),ESF(INN,12),AST(INN,2),R(INN)
INTEGER IP(12),SE
INTEGER DE(INN),JDES
REAL KE(INN),DP(12),DB(12)
REAL PO(NX),PL(12),PG(12),P1(INN,12)
CHARACTER TIPO(INN) *I,CLASS *I,CA *3

COMMON /R7/ ESF,LDELTD,AST,ID,PP,JDES,L,NF,ASMAX,ASMIN,ASMED,TMIN,
TMAX

2000 OPEN (1,FILE='DP_NOS.TXT',STATUS='OLD')
OPEN (2,FILE='DP_BAR.TXT',STATUS='OLD')
OPEN (3,FILE='DP_APOIO.TXT',STATUS='OLD')
OPEN (4,FILE='DP_CN.TXT',STATUS='OLD')
OPEN (5,FILE='DP_CB.TXT',STATUS='OLD')
OPEN (6,FILE='ESFORÇOS.TXT',STATUS='NEW')
OPEN (7,FILE='DESLO.TXT',STATUS='NEW')
OPEN (8,FILE='DP_PARA.TXT',STATUS='OLD')
OPEN (9,FILE='DP_SEC.TXT',STATUS='OLD')
OPEN (10,FILE='ARMADURA.TXT',STATUS='NEW')

*
2100 LEITURA DOS PARAMETROS DE CALCULO
READ(8,*) E,LDELTD,FCK,GAMAC,GMCD,CA,IACO,CLASS,FYK,GAMAS,
GMSD,TMAX,TMIN
TMAX=TMAX/100.
TMIN=TMIN/100.
ES=2100000000.
DFS=0.

*
2150 COORDENADAS DOS PONTOS
GOTO 10000

```

```

*      INCIDENCIA DAS BARRAS CARAC. GEOM. DAS SECOES TRANSV. DAS BARRAS
2160  GOTO 12000

*      CALCULO DOS COMPRIMENTOS E INCLINACOES
2180  GOTO 14000

*      CALC. DA MATRIZ DE RIGIDEZ DA ESTRUTURA
2190  GOTO 20000

*      LEITURA DAS CARGAS NODAIS
2200  GOTO 22000

*      CORREÇÃO DEVIDO AOS RECALQUES DE APOIO
2205  GOTO 23000

*      CARGAS NAS BARRAS
2207  GOTO 23700

*      RESOLUCAO DO SISTEMA DE EQUAÇÕES (DETERM. DE D)
2210  GOTO 24000

*      CÁLCULO DOS ESFORÇOS FINAIS NAS BARRAS
2220  GOTO 26000

*      CÁLCULO DAS REAÇÕES DE APOIO
2230  GOTO 28000

*      CÁLCULO DAS REAÇÕES NAS MOLAS
2240  GOTO 29100

*      COORDENADAS DOS PONTOS
10000 NP=0
      NP1=0

      DO 10151 II=1,999
      READ(1,*,END=12000) I,X(I),Y(I),Z(I)
      WRITE(7,*) I,X(I),Y(I),Z(I)
      NP=NP+1
      NP1=NP+1
10151 CONTINUE
      GOTO 2160

*      INCIDENCIAS DAS BARRAS
12000 NB=0
      DO 12153 II=1,999
      READ(2,*,END=2180) I,NI(I),NF(I),ASEC(I),JX(I),JY(I),
*      NSECAO(I),TIPO(I)
      WRITE(7,*) I,NI(I),NF(I),ASEC(I),JX(I),JY(I),',',NSECAO(I)
*      ,TIPO(I)
      NB=NB+1
12153 CONTINUE
      GOTO 2180

*      CALCULO DE COMPRIMENTOS E INCLINACOES

14000 PI=3.141592654

      IF (LDELTEQ.1) GOTO 14035
      DO 14162 I=1,NB

      DELTAX=(X(NF(I))-X(NI(I)))/LDELTE
      DELTAY=(Y(NF(I))-Y(NI(I)))/LDELTE
      DELTAZ=(Z(NF(I))-Z(NI(I)))/LDELTE
      DO 14160 JJ=1,LDELTE-1
      X(NP+1)=X(NI(I))+DELTAX*JJ
      Y(NP+1)=Y(NI(I))+DELTAY*JJ
      Z(NP+1)=Z(NI(I))+DELTAZ*JJ
      NP=NP+1
      WRITE(7,*) NP,X(NP),Y(NP),Z(NP)
14160 CONTINUE
14162 CONTINUE

      DO 14165 I=NB,I,-1
      NI((I-1)*LDELTE+1)=NI(I)
      NF(1*LDELTE)=NF(I)
14165 CONTINUE

      NP2=NP-(NB*(LDELTE-1))

      DO 14167 I=1,NB
      NF((I-1)*LDELTE+1)=NP2+1
      NP2=NP2+1

      DO 14166 JJ=2,LDELTE-1
      NI((I-1)*LDELTE+JJ)=NP2
      NF((I-1)*LDELTE+JJ)=NP2+1
      NP2=NP2+1
14166 CONTINUE
      NI((I-1)*LDELTE+JJ)=NP2

14167 CONTINUE

```

```

NB1=NB
NB=NB*LDELTA

DO 14168 I=NB,1,-1
  IF (1.0*I/LDELTA.EQ.INT(I/LDELTA)) THEN
    NSECAO(I)=NSECAO(INT(I/LDELTA))
    ASEC(I)=ASEC(INT(I/LDELTA))
    JX(I)=JX(INT(I/LDELTA))
    JY(I)=JY(INT(I/LDELTA))
  ELSE
    NSECAO(I)=NSECAO(INT(I/LDELTA)+1)
    ASEC(I)=ASEC(INT(I/LDELTA)+1)
    JX(I)=JX(INT(I/LDELTA)+1)
    JY(I)=JY(INT(I/LDELTA)+1)
  ENDIF
14168 CONTINUE

14035 DO 14150 I=1,NB
14040   DX=X(NF(I))-X(NI(I))
14050   DY=Y(NF(I))-Y(NI(I))
14055   DZ=Z(NF(I))-Z(NI(I))
14060   L(I)=SQRT(DX*DX+DY*DY+DZ*DZ)
14080   CX(I)=DX/L(I)
        IF (CX(I).EQ.0.0) CX(I)=1.0E-18
14090   CY(I)=DY/L(I)
        IF (CY(I).EQ.0.0) CY(I)=1.0E-18
14100   CZ(I)=DZ/L(I)
        IF (CZ(I).EQ.0.0) CZ(I)=1.0E-18
14150 CONTINUE
14170 GOTO 2190

*   CALCULO DA MATRIZ DE RIGIDEZ DA ESTRUTURA

*   GRAU DE LIBERDADE (NÓ DE BARRA)
20000 ND=6
20005 NL=12
*   DEFINICAO DAS INCOGNITAS
20020 GOTO 20180

*   ZERA MATRIZ K
20030 GOTO 20300

20040 DO 20150 IJ=1,NB
20045 I=IJ

*   CALCULO DA MATRIZ KB DA BARRA I (SLR)

V=E*ASEC(I)/L(I)
KB(1,1)=V
KB(7,7)=V
KB(1,7)=-V
KB(7,1)=-V

V=12.0*E*JX(I)/L(I)/L(I)/L(I)
KB(2,2)=V
KB(8,8)=V
KB(2,8)=-V
KB(8,2)=-V

V=6.0*E*JX(I)/L(I)/L(I)
KB(2,6)=V
KB(6,2)=V
KB(2,12)=V
KB(12,2)=V
KB(6,8)=-V
KB(8,6)=-V
KB(8,12)=-V
KB(12,8)=-V

KB(12,12)=V
KB(6,12)=V/2.0
KB(12,6)=V/2.0

V=12.0*E*JY(I)/L(I)/L(I)/L(I)
KB(3,3)=V
KB(9,9)=V
KB(3,9)=-V
KB(9,3)=-V

V=6.0*E*JY(I)/L(I)/L(I)
KB(3,5)=-V
KB(5,3)=-V
KB(3,11)=-V
KB(11,3)=-V
KB(5,9)=V
KB(9,5)=V
KB(9,11)=V
KB(11,9)=V

V=4.0*E*JY(I)/L(I)
KB(5,5)=V
KB(11,11)=V
KB(5,11)=V/2.0
KB(11,5)=V/2.0

```

```

KB(4,4)=1.0
KB(10,10)=1.0
KB(4,10)=-1.0
KB(10,4)=-1.0

```

* CALCULO DA MATRIZ T DA BARRA I

```

20088 DO 20092 JJ=1,NL
20089     DO 20091 K=1,NL
20090         T(JJ,K)=0.0
20091     CONTINUE
20092 CONTINUE

```

```

IF (CY(I).EQ.1.0.OR.CY(I).EQ.-1.0) THEN
    CB=0.707106781
    SB=CB*CY(I)
ELSE

```

```

    DY=Y(NF(I))-PP(ID(NF(I),2))-(Y(NI(I))-PP(ID(NI(I),2)))
    DX=X(NF(I))-PP(ID(NF(I),3))-(X(NI(I))-PP(ID(NI(I),3)))
    DZ=Z(NF(I))-PP(ID(NF(I),1))-(Z(NI(I))-PP(ID(NI(I),1)))
    AL=SQRT(DX*DX+DY*DY+DZ*DZ)
    ACY=DY/AL

```

```

IF (ACY.EQ.1.0.OR.ACY.EQ.-1.0) THEN
    DX1=PP(ID(NF(I),3))
    DZ1=PP(ID(NF(I),1))
    AL1=SQRT(DX1*DX1+DZ1*DZ1)
    CB=DZ1/AL1
    SB=DX1/AL1

```

```

IF (CB.EQ.0.) THEN
    ANG=PI
ELSE

```

```

    ANG=ATAN(SB/CB)
ENDIF

```

```

IF (CB.GT.0.) THEN
    IF (SB.GE.0.) THEN
        SB=SIN(PI-ANG)
        CB=-COS(PI-ANG)
    ELSE

```

```

        SB=SIN(ANG)
        CB=COS(ANG)
    ENDIF

```

```

ELSE
    IF (SB.LT.0.) THEN
        SB=SIN(2.0*PI-ANG)
        CB=-COS(2.0*PI-ANG)
    ELSE

```

```

        SB=SIN(PI+ANG)
        CB=COS(PI+ANG)
    ENDIF

```

```

ENDIF
ELSE
    CB=1.0
    SB=0.0
ENDIF

```

```

ENDIF
T(1,1)=CZ(I)
T(1,2)=CY(I)
T(1,3)=CX(I)
T(2,1)=(-CZ(I)*CY(I)*CB-CX(I)*SB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))
T(2,2)=SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))*CB
T(2,3)=(CZ(I)*SB-CY(I)*CX(I)*CB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))
T(3,1)=(CZ(I)*CY(I)*SB-CX(I)*CB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))
T(3,2)=-1.0*SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))*SB
T(3,3)=(CY(I)*CX(I)*SB+CZ(I)*CB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))

T(4,4)=CZ(I)
T(4,5)=CY(I)
T(4,6)=CX(I)
T(5,4)=(-CZ(I)*CY(I)*CB-CX(I)*SB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))
T(5,5)=SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))*CB
T(5,6)=(CZ(I)*SB-CY(I)*CX(I)*CB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))
T(6,4)=(CZ(I)*CY(I)*SB-CX(I)*CB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))
T(6,5)=-1.0*SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))*SB
T(6,6)=(CY(I)*CX(I)*SB+CZ(I)*CB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))

T(7,7)=CZ(I)
T(7,8)=CY(I)
T(7,9)=CX(I)
T(8,7)=(-CZ(I)*CY(I)*CB-CX(I)*SB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))
T(8,8)=SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))*CB
T(8,9)=(CZ(I)*SB-CY(I)*CX(I)*CB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))
T(9,7)=(CZ(I)*CY(I)*SB-CX(I)*CB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))
T(9,8)=-1.0*SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))*SB
T(9,9)=(CY(I)*CX(I)*SB+CZ(I)*CB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))

T(10,10)=CZ(I)
T(10,11)=CY(I)
T(10,12)=CX(I)
T(11,10)=(-CZ(I)*CY(I)*CB-CX(I)*SB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))
T(11,11)=SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))*CB
T(11,12)=(CZ(I)*SB-CY(I)*CX(I)*CB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))
T(12,10)=(CZ(I)*CY(I)*SB-CX(I)*CB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))
T(12,11)=-1.0*SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))*SB
T(12,12)=(CY(I)*CX(I)*SB+CZ(I)*CB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))

```



```

*      CALCULO DA MATRIZ KP=T(T)*KB*T - BARRA I
20107 DO 20115 II=1,NL
20108   DO 20114 JJ=1,NL
20109     S=0.0
20110     DO 20112 K=1,NL
20111       S=S+T(K,II)*KB(K,JJ)
20112     CONTINUE
20113     SS(II,JJ)=S
20114   CONTINUE
20115 CONTINUE
20116 DO 20124 II=1,NL
20117   DO 20123 JJ=1,NL
20118     S=0.0
20119     DO 20121 K=1,NL
20120       S=S+SS(II,K)*T(K,JJ)
20121     CONTINUE
20122     KP(II,JJ)=S
20123   CONTINUE
20124 CONTINUE

*      SOMA KP EM K
20127 DO 20130 II=1,ND
20128   IP(II)=ID(NI(II),II)
20129   IP(II+ND)=ID(NF(II),II)
20130 CONTINUE
20131 DO 20140 II=1,NL
20132   IF (IP(II)) 20140,20140,20133
20133   DO 20136 JJ=1,NL
20134     IF (IP(JJ)) 20136,20136,20135
20135     KK(IP(II),IP(JJ))=KK(IP(II),IP(JJ))+KP(II,JJ)
20136   CONTINUE
20140 CONTINUE
20150 CONTINUE
20160 GOTO 2200

*      DEFINICAO DAS INCOGNITAS
20180 REWIND 3
20185 DO 20215 I=NP*ND,1,-1
20190   READ(3,*,END=20225) II,JJ,P(I)
20190   ID(II,JJ)=I
20215 CONTINUE
20225 JJ=1
20225   N=ID(II,JJ)-I
20230 DO 20270 II=1,NP
20240   DO 20260 JJ=1,ND
20257     IF (ID(II,JJ).EQ.0) THEN
20257       ID(II,JJ)=JJ
20257       JJ=JJ+1
20257     ENDIF
20260 CONTINUE
20270 CONTINUE
20280 GOTO 20030

*      ZERA A MATRIZ KK
20300 DO 20360 II=1,N
20330   DO 20350 JJ=1,N
20340     KK(II,JJ)=0.0
20350   CONTINUE
20360 CONTINUE
20370 GOTO 20040

*      LEITURA DAS CARGAS NODAIS
22000 DO 22040 I=1,N
22030   P(I)=0.
22040 CONTINUE
22070 DO 22090 I=1,NP*ND
22080   READ(4,*,END=22110) NC,JJ,P(ID(NC,JJ))
22090 CONTINUE
22110 GOTO 2205

*      EFEITO DOS RECALQUES DE APOIO
23000 DO 23060 I=1,N
23030   DO 23050 JJ=N+1,ND*NP
23040     P(I)=P(I)-KK(I,JJ)*P(JJ)
23050   CONTINUE
23060 CONTINUE
23070 GOTO 2207

*      CARGAS NAS BARRAS
23700 II=0
23700 DO 23705 JJ=1,NL
23700   PL(JJ)=0.
23705 CONTINUE
LT=0

```

```

23710 DO 23985 JB=1,10000

      IF (LT.EQ 0) THEN
        READ (5,*,END=23950) I,DRT,A1,C,P2
        LDELTI=L(I*LDELTI)
        I1=INT(A1/LDELTI)
        I=(I-1)*LDELTI+I1+1
        AC1=1.0*I1
        A1=A1-AC1*LDELTI
        C=C-AC1*LDELTI
        IF ((C-A1).LE.LDELTI) THEN
          LT=0
        ELSE
          LT=1
          C1=C-LDELTI
          C=LDELTI-A1
          I1=I
        ENDIF
      ELSE
        I=I1+1
        I1=I
        A1=0.
        IF (C1.LE.LDELTI) THEN
          C=C1
          LT=0
        ELSE
          C=LDELTI
          C1=C1-LDELTI
        ENDIF
      ENDIF
      IF (I1.EQ.0) I1=1
      IF (I.EQ.11) GOTO 23750
      GOTO 23950
23742 DO 23748 JJ=1,NL
23745   PL(JJ)=0.
23748 CONTINUE
23750 IF (A1.NE.C) GOTO 23900

*
      CARGA CONCENTRADA

23855 L1=L(1)
23862 B1=L1-A1
23863 PA=-P2*A1*B1*B1/L1/L1
23864 PB1=P2*A1*A1*B1/L1/L1
      IF (DRT.EQ.2) THEN
        PL(6)=PL(6)+PA
        PL(12)=PL(12)+PB1
        PL(2)=PL(2)-P2*B1/L1+(PA+PB1)/L1
        PL(8)=PL(8)-P2*A1/L1-(PA+PB1)/L1
      ELSEIF (DRT.EQ.3) THEN
        PL(5)=PL(5)-PA
        PL(11)=PL(11)-PB1
        PL(3)=PL(3)-P2*B1/L1+(PA+PB1)/L1
        PL(9)=PL(9)-P2*A1/L1-(PA+PB1)/L1
      ENDIF
23870 I=0
      GOTO 23985

*
      CARGA DISTRIBUIDA

23900 L1=L(1)
23930 AB=A1+C/2
23932 BB=L1-AB
23934 PA=-P2*L1*L1/12*C/L1*(12*AB/L1*BB*BB/L1/L1-C*C/L1/L1*(3*BB/L1-1))
23936 PB1=P2*L1*L1/12*C/L1*(12*AB*AB/L1/L1*BB/L1-C*C/L1/L1*(3*AB/L1-1))
      IF (DRT.EQ.2) THEN
        PL(6)=PL(6)+PA
        PL(12)=PL(12)+PB1
        PL(2)=PL(2)-P2*C*BB/L1+(PA+PB1)/L1
        PL(8)=PL(8)-P2*C*AB/L1-(PA+PB1)/L1
      ELSEIF (DRT.EQ.3) THEN
        PL(5)=PL(5)-PA
        PL(11)=PL(11)-PB1
        PL(3)=PL(3)-P2*C*BB/L1+(PA+PB1)/L1
        PL(9)=PL(9)-P2*C*AB/L1-(PA+PB1)/L1
      ENDIF
23946 I=0
      GOTO 23985

23950 IF (I1.EQ.0) GOTO 2210
      DO 23953 JJ=1,ND
23951   IP(JJ)=ID(NI(I1),JJ)
23952   IP(JJ+ND)=ID(NF(I1),JJ)
23953 CONTINUE

*
      CALCULO DE T
23954 DO 23958 JJ=1,NL
23955   DO 23957 K=1,NL
23956     T(JJ,K)=0.0
23957   CONTINUE
23958 CONTINUE
      I2=I
      I=I1

```

```

IF (CY(I).EQ.1.0.OR.CY(I).EQ.-1.0) THEN
  CB=0.707106781
  SB=CB*CY(I)
ELSE
  DY=Y(NF(I))-PP(ID(NF(I),2))-(Y(NI(I))-PP(ID(NI(I),2)))
  DX=X(NF(I))-PP(ID(NF(I),3))-(X(NI(I))-PP(ID(NI(I),3)))
  DZ=Z(NF(I))-PP(ID(NF(I),1))-(Z(NI(I))-PP(ID(NI(I),1)))
  AL=SQRT(DX*DX+DY*DY+DZ*DZ)
  ACY=DY/AL

  IF (ACY.EQ.1.0.OR.ACY.EQ.-1.0) THEN
    DX1=PP(ID(NF(I),3))
    DZ1=PP(ID(NF(I),1))
    AL1=SQRT(DX1*DX1+DZ1*DZ1)
    CB=DZ1/AL1
    SB=DX1/AL1

    IF (CB.EQ.0.) THEN
      ANG=PI
    ELSE
      ANG=ATAN(SB/CB)
    ENDIF

    IF (CB.GT.0.0) THEN
      IF (SB.GE.0.) THEN
        SB=SIN(PI-ANG)
        CB=-COS(PI-ANG)
      ELSE
        SB=SIN(ANG)
        CB=COS(ANG)
      ENDIF
    ELSE
      IF (SB.LT.0.) THEN
        SB=SIN(2.0*PI-ANG)
        CB=-COS(2.0*PI-ANG)
      ELSE
        SB=SIN(PI+ANG)
        CB=COS(PI+ANG)
      ENDIF
    ENDIF
  ELSE
    CB=1.0
    SB=0.0
  ENDIF
ENDIF

T(1,1)=CZ(I)
T(1,2)=CY(I)
T(1,3)=CX(I)
T(2,1)=(-CZ(I)*CY(I)*CB-CX(I)*SB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))
T(2,2)=SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))*CB
T(2,3)=(CZ(I)*SB-CY(I)*CX(I)*CB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))
T(3,1)=(CZ(I)*CY(I)*SB-CX(I)*CB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))
T(3,2)=-1.0*SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))*SB
T(3,3)=(CY(I)*CX(I)*SB+CZ(I)*CB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))

T(4,4)=CZ(I)
T(4,5)=CY(I)
T(4,6)=CX(I)
T(5,4)=(-CZ(I)*CY(I)*CB-CX(I)*SB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))
T(5,5)=SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))*CB
T(5,6)=(CZ(I)*SB-CY(I)*CX(I)*CB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))
T(6,4)=(CZ(I)*CY(I)*SB-CX(I)*CB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))
T(6,5)=-1.0*SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))*SB
T(6,6)=(CY(I)*CX(I)*SB+CZ(I)*CB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))

T(7,7)=CZ(I)
T(7,8)=CY(I)
T(7,9)=CX(I)
T(8,7)=(-CZ(I)*CY(I)*CB-CX(I)*SB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))
T(8,8)=SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))*CB
T(8,9)=(CZ(I)*SB-CY(I)*CX(I)*CB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))
T(9,7)=(CZ(I)*CY(I)*SB-CX(I)*CB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))
T(9,8)=-1.0*SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))*SB
T(9,9)=(CY(I)*CX(I)*SB+CZ(I)*CB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))

T(10,10)=CZ(I)
T(10,11)=CY(I)
T(10,12)=CX(I)
T(11,10)=(-CZ(I)*CY(I)*CB-CX(I)*SB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))
T(11,11)=SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))*CB
T(11,12)=(CZ(I)*SB-CY(I)*CX(I)*CB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))
T(12,10)=(CZ(I)*CY(I)*SB-CX(I)*CB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))
T(12,11)=-1.0*SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))*SB
T(12,12)=(CY(I)*CX(I)*SB+CZ(I)*CB)/SQRT(CZ(I)*CZ(I)+CX(I)*CX(I))

I=12

23960 DO 23972 JJ=1,NL
23962   S=0.0
23964   DO 23968 K=1,NL
23966     S=S+T(K,JJ)*PL(K)
23968   CONTINUE
23970   PG(JJ)=S
23972 CONTINUE

```

```

23974 DO 23982 JJ=1,NL
23976 IF (IP(JJ).EQ.0) GOTO 23981
23978 PO(IP(JJ))=PO(IP(JJ))+PG(JJ)
23980 P1(I1,JJ)=PL(JJ)
23982 CONTINUE
23983 IF (I.EQ.0) GOTO 23990
      II=I
      GOTO 23742
23985 CONTINUE

23990 DO 23996 I=1,N
23994 P(I)=P(I)-PO(I)
23996 CONTINUE
23997 GOTO 2210

```

* RESOLUCAO DE SISTEMA DE EQUACOES LINEARES

* TRIANGULARIZACAO

```

24000 DO 24180 I=1,N
24050 V=KK(I,I)
24060 DO 24080 JJ=1,N
24070 KK(I,JJ)=KK(I,JJ)/V
24080 CONTINUE
24090 P(I)=P(I)/V

24100 IF (I.EQ.N) GOTO 24200
24110 DO 24170 JJ=I+1,N
24120 V=KK(JJ,I)
24130 DO 24150 K=I,N
24140 KK(JJ,K)=KK(JJ,K)-V*KK(I,K)
24150 CONTINUE
24160 P(JJ)=P(JJ)-V*P(I)
24170 CONTINUE
24180 CONTINUE

```

* RETROSUBSTITUICAO

```

24200 DO 24260 II=N-1,1,-1
24230 DO 24250 JJ=II+1,N
24240 P(II)=P(II)-KK(II,JJ)*P(JJ)
24250 CONTINUE
24260 CONTINUE

24410 GOTO 2220

```

* CALCULO DOS ESFORCOS FINAIS NAS BARRAS

```

26000 LT=1
26010 IESF = 0
26070 DO 26330 I=1,NB
26090 DO 26120 II=1,ND
26100 IP(II)=ID(NI(I),II)
26110 IP(II+ND)=ID(NF(I),II)
26120 CONTINUE
26130 DO 26150 II=1,NL
26140 DP(II)=P(IP(II))
26150 CONTINUE

```

* CALCULO DE T

```

26152 DO 26158 JJ=1,NL
26154 DO 26157 K=1,NL
26156 T(JJ,K)=0.
26157 CONTINUE
26158 CONTINUE

IF (CY(I).EQ.1.0 OR CY(I).EQ.-1.0) THEN
  CB=0.707106781
  SB=CB*CY(I)
ELSE
  DY=Y(NF(I))-PP(ID(NF(I),2))-(Y(NI(I))-PP(ID(NI(I),2)))
  DX=X(NF(I))-PP(ID(NF(I),3))-(X(NI(I))-PP(ID(NI(I),3)))
  DZ=Z(NF(I))-PP(ID(NF(I),1))-(Z(NI(I))-PP(ID(NI(I),1)))
  AL=SQRT(DX*DX+DY*DY+DZ*DZ)
  ACY=DY/AL

  IF (ACY.EQ.1.0 OR ACY.EQ.-1.0) THEN
    DX1=PP(ID(NF(I),3))
    DZ1=PP(ID(NF(I),1))
    AL1=SQRT(DX1*DX1+DZ1*DZ1)
    CB=DZ1/AL1
    SB=DX1/AL1
    IF (CB.EQ.0.) THEN
      ANG=PI
    ELSE
      ANG=ATAN(SB/CB)
    ENDIF
    IF (CB.GT.0.0) THEN
      IF (SB.GE.0.) THEN
        SB=SIN(PI-ANG)
        CB=-COS(PI-ANG)
      ELSE

```

```

        SB=SIN(ANG)
        CB=cos(ANG)
    ENDIF
ELSE
    IF (SB.LT.0.) THEN
        SB=SIN(2.0*PI-ANG)
        CB=-COS(2.0*PI-ANG)
    ELSE
        SB=SIN(PI+ANG)
        CB=COS(PI+ANG)
    ENDIF
ENDIF
ELSE
    CB=1.0
    SB=0.0
ENDIF
ENDIF
T(1,1)=CZ(1)
T(1,2)=CY(1)
T(1,3)=CX(1)
T(2,1)=-CZ(1)*CY(1)*CB-CX(1)*SB/SQRT(CZ(1)*CZ(1)+CX(1)*CX(1))
T(2,2)=SQRT(CZ(1)*CZ(1)+CX(1)*CX(1))*CB
T(2,3)=(CZ(1)*SB-CY(1)*CX(1)*CB)/SQRT(CZ(1)*CZ(1)+CX(1)*CX(1))
T(3,1)=(CZ(1)*CY(1)*SB-CX(1)*CB)/SQRT(CZ(1)*CZ(1)+CX(1)*CX(1))
T(3,2)=-1.0*SQRT(CZ(1)*CZ(1)+CX(1)*CX(1))*SB
T(3,3)=(CY(1)*CX(1)*SB+CZ(1)*CB)/SQRT(CZ(1)*CZ(1)+CX(1)*CX(1))

T(4,4)=CZ(1)
T(4,5)=CY(1)
T(4,6)=CX(1)
T(5,4)=-CZ(1)*CY(1)*CB-CX(1)*SB/SQRT(CZ(1)*CZ(1)+CX(1)*CX(1))
T(5,5)=SQRT(CZ(1)*CZ(1)+CX(1)*CX(1))*CB
T(5,6)=(CZ(1)*SB-CY(1)*CX(1)*CB)/SQRT(CZ(1)*CZ(1)+CX(1)*CX(1))
T(6,4)=(CZ(1)*CY(1)*SB-CX(1)*CB)/SQRT(CZ(1)*CZ(1)+CX(1)*CX(1))
T(6,5)=-1.0*SQRT(CZ(1)*CZ(1)+CX(1)*CX(1))*SB
T(6,6)=(CY(1)*CX(1)*SB+CZ(1)*CB)/SQRT(CZ(1)*CZ(1)+CX(1)*CX(1))

T(7,7)=CZ(1)
T(7,8)=CY(1)
T(7,9)=CX(1)
T(8,7)=-CZ(1)*CY(1)*CB-CX(1)*SB/SQRT(CZ(1)*CZ(1)+CX(1)*CX(1))
T(8,8)=SQRT(CZ(1)*CZ(1)+CX(1)*CX(1))*CB
T(8,9)=(CZ(1)*SB-CY(1)*CX(1)*CB)/SQRT(CZ(1)*CZ(1)+CX(1)*CX(1))
T(9,7)=(CZ(1)*CY(1)*SB-CX(1)*CB)/SQRT(CZ(1)*CZ(1)+CX(1)*CX(1))
T(9,8)=-1.0*SQRT(CZ(1)*CZ(1)+CX(1)*CX(1))*SB
T(9,9)=(CY(1)*CX(1)*SB+CZ(1)*CB)/SQRT(CZ(1)*CZ(1)+CX(1)*CX(1))

T(10,10)=CZ(1)
T(10,11)=CY(1)
T(10,12)=CX(1)
T(11,10)=-CZ(1)*CY(1)*CB-CX(1)*SB/SQRT(CZ(1)*CZ(1)+CX(1)*CX(1))
T(11,11)=SQRT(CZ(1)*CZ(1)+CX(1)*CX(1))*CB
T(11,12)=(CZ(1)*SB-CY(1)*CX(1)*CB)/SQRT(CZ(1)*CZ(1)+CX(1)*CX(1))
T(12,10)=(CZ(1)*CY(1)*SB-CX(1)*CB)/SQRT(CZ(1)*CZ(1)+CX(1)*CX(1))
T(12,11)=-1.0*SQRT(CZ(1)*CZ(1)+CX(1)*CX(1))*SB
T(12,12)=(CY(1)*CX(1)*SB+CZ(1)*CB)/SQRT(CZ(1)*CZ(1)+CX(1)*CX(1))

26170 DO 26230 II=1,NL
26180     S=0.0
26190     DO 26210 JJ=1,NL
26200         S=S+T(II,JJ)*DP(JJ)
26210     CONTINUE
26220     DB(II)=S
26230 CONTINUE

*   CALCULO DA MATRIZ KB DA BARRA I (SLR)

V=E*ASEC(1)/L(1)
KB(1,1)=V
KB(7,7)=-V
KB(1,7)=-V
KB(7,1)=-V

V=12.0*E*JX(1)/L(1)/L(1)
KB(2,2)=V
KB(8,8)=V
KB(2,8)=-V
KB(8,2)=-V

V=6.0*E*JX(1)/L(1)/L(1)
KB(2,6)=V
KB(6,2)=V
KB(2,12)=V
KB(12,2)=V
KB(6,8)=-V
KB(8,6)=-V
KB(8,12)=-V
KB(12,8)=-V

V=4.0*E*JX(1)/L(1)
KB(6,6)=V
KB(12,12)=V
KB(6,12)=V/2.0
KB(12,6)=V/2.0

```

```

V=12.0*E*JY(I)/L(I)/L(I)/L(I)
KB(3,3)=V
KB(9,9)=V
KB(3,9)=-V
KB(9,3)=-V

V=6.0*E*JY(I)/L(I)/L(I)
KB(3,5)=-V
KB(5,3)=-V
KB(3,11)=-V
KB(11,3)=-V
KB(5,9)=V
KB(9,5)=V
KB(9,11)=V
KB(11,9)=V

V=4.0*E*JY(I)/L(I)
KB(5,5)=V
KB(11,11)=V
KB(5,11)=V/2.0
KB(11,5)=V/2.0

KB(4,4)=1.0
KB(4,10)=-1.0
KB(10,4)=-1.0

26250 DO 26320 I=1,NL
26260   S=P1(I,I)
26270   DO 26290 JJ=1,NL
26280     S=S+KB(I,JJ)*DB(JJ)
26290   CONTINUE
26300   PB(I)=S
26310   ESF(I,I)=PB(I)
26320 CONTINUE

26330 CONTINUE
JDES=0
DO 26335 I=1,NP
  X(I)=X(I)-PP(ID(I,3))
  Z(I)=Z(I)-PP(ID(I,1))
26335 CONTINUE

* CALL DIMSEC(E,NSECAO,FCK,GAMAC,GMCD,CA,IACO,CLASS,ES,FYK,
  GAMAS,GMSD,DFS,NB)

26340 GOTO 2230

* CALCULO DAS REACOES DE APOIO

28000 DO 28090 I=N+1,ND*NP
28040   R(I)=PO(I)
28050   DO 28070 JJ=1,ND*NP
28060     R(I)=R(I)+KK(I,JJ)*P(JJ)
28070   CONTINUE
28090 CONTINUE
REWIND 3
DO 28105 I=1,N
  READ(3,*,END=28110) I1,I2,S
28105 CONTINUE
28110 GOTO 2240

29100 IF (JDES.EQ.1) THEN
  GOTO 29200
ELSE
  DO 29115 I=1,NP
    PP(ID(I,3))=0
    PP(ID(I,1))=0
29115 CONTINUE
ENDIF

29125 DO 29130 I=1,ND*NP
  DO 29130 JJ=1,ND*NP
    KK(I,JJ)=0.
29130 CONTINUE
29131 DO 29132 I=1,NL
  PL(I)=0.
29132 CONTINUE
DO 29140 I=1,ND*NP
  P(I)=0.
  PO(I)=0.
29140 CONTINUE
REWIND 1
REWIND 2
REWIND 3
REWIND 4
REWIND 5
REWIND 8
DO 29155 I=1,NP
  X(I)=X(I)+PP(ID(I,3))
  Z(I)=Z(I)+PP(ID(I,1))
29155 CONTINUE

GO TO 14030
29200 END

```

```

SUBROUTINE ACO1(CURVA,BETA,FNS,AMUS,AMVS)
COMMON /R1/ NSEC,NV,NA,NP,NCIRC(20),NEXT(100),X(10),Y(10),
* INCI(2,10),BITO(2),CORDX(10),CORDY(10),IP,JP,BETAS(9),AREF,
* CURVAS(9)
COMMON /R2/ FCK,GAMAC,IACO,ZCLAS,ES,FYK,GAMAS,DTETA,TETA1,BETA1,
1 TOLE, FNORM,NBT,NFT,NFTD,AREAC, FNMAX,AMXLI,AMYLI, FNMIN,AMXMS,
2 AMYMS, FNMC,SGCD,FYD,EYDA,EYDB,EPD,AKEY, DBETA,AMOMX,AMOMY,
3 DFS,CURVAI
COMMON /R8/ GMCD,GMSD
COMMON /R3/ U(100),V(100),CORDU(200),CORDV(200),VMAX,VMIN,VSMIN,H,
ID,TETAS(9),FW(10)

A9=CURVA
B9=-CURVA*(VMAX-BETA*D)

40 FNS=0.
AMUS=0.
AMVS=0.
DO 50 I=1,NBT
EPS=A9*CORDV(I)+B9
SIG=SIGMA(EPS)
FNS=FNS+BITO(2)*SIG
AMUS=AMUS+BITO(2)*SIG*CORDV(I)
AMVS=AMVS+BITO(2)*SIG*CORDU(I)
50 CONTINUE
RETURN
END

SUBROUTINE CARDS
C LE OS DADOS DO PROBLEMA.
COMMON /R1/ NSEC,NV,NA,NP,NCIRC(20),NEXT(100),X(10),Y(10),
* INCI(2,10),BITO(2),CORDX(10),CORDY(10),IP,JP,BETAS(9),AREF,
* CURVAS(9)
COMMON /R2/ FCK,GAMAC,IACO,ZCLAS,ES,FYK,GAMAS,DTETA,TETA1,BETA1,
1 TOLE, FNORM,NBT,NFT,NFTD,AREAC, FNMAX,AMXLI,AMYLI, FNMIN,AMXMS,
2 AMYMS, FNMC,SGCD,FYD,EYDA,EYDB,EPD,AKEY, DBETA,AMOMX,AMOMY,
3 DFS,CURVAI
COMMON /R8/ GMCD,GMSD
DATA VERT/'VERT',POLI/'POLI',ARMA/'ARMA',CARA/'CARA'
DATA TETA1/'TETA',FNOR/'FNOR',TOLA/'TOLE',BRANC' '/'
DATA AINFO/'INFO'
NBT=0
NFT=0

READ(IP,*) NV,NA,NP,(NCIRC(I),I=1,NP)
DO 10 I=1,NV
READ(IP,*) NEXT(I),X(I),Y(I)
10 CONTINUE
L=0
DO 20 I=1,NP
K=NCIRC(I)
DO 20 J=1,K
L=L+1
INCI(I,J)=L
20 CONTINUE
DO 25 I=1,NA
READ(IP,*) NI,CORDX(NI),CORDY(NI)
25 CONTINUE
NBT=NA
REWIND 9
RETURN
STOP
END

SUBROUTINE CIRC(X,Y,N,A,AMXX,AMYY)
C DADO UM POLIGONO ATRAVES DAS COORDENADAS DE SEUS VERTICES NUM
C SISTEMA OXY, REUNIDAS NOS VETORES X E Y DE N ELEMENTOS, A
C SUBROTINA CALCULA SUA AREA E SEUS MOMENTOS ESTATICOS AMXX E AMYY
C RELATIVAMENTE A ESSES EIXOS.

DIMENSION X(1),Y(1)
IF(N.EQ.0) GO TO 20
X(N+1)=X(1)
Y(N+1)=Y(1)
A=0.
AMXX=0.
AMYY=0.
DO 10 K=1,N
AUX=X(K)*Y(K+1)-X(K+1)*Y(K)
A=A+AUX
AMXX=AMXX+(Y(K)+Y(K+1))*AUX
AMYY=AMYY+(X(K)+X(K+1))*AUX
10 CONTINUE
A=.5*A
AMXX=AMXX/6.
AMYY=AMYY/6.

```

```

20   RETURN
      A=0.
      AMXX=0.
      AMYY=0.
      RETURN
      END

SUBROUTINE CONCI(CURVA,BETA,FNC,AMUC,AMVC)

COMMON /R2/ FCK,GAMAC,IACO,ZCLAS,ES,FYK,GAMAS,DTETA,TETA1,BETA1,
1  TOLE,FNORM,NBT,NFT,NFTD,AREAC,FNMAX,AMXLI,AMyli,FNMIN,AMXMS,
2  AMYMS,FNMC,SGCD,FYD,EYDA,EYDB,EPD,AKEY,DBETA,AMOMX,AMOMY,
3  DFS,CURVA1
COMMON /R8/ GMCD,GMSD
COMMON /R3/ U(100),V(100),CORDU(200),CORDV(200),VMAX,VMIN,VSMIN,H,
ID,TETAS(9),FW(10)

      FNC=0.
      AMUC=0.
      AMVC=0.
      EPSI = CURVA*BETA*D

20   IF (EPSI .E. 0) GO TO 30
      SGCD=SGCD*GAMAC/GMCD
      CALL SEPAC2(BETA,FNC,AMUC,AMVC,CURVA)
      SGCD=SGCD*GMCD/GAMAC
30   RETURN
      END

SUBROUTINE CONC2(BETA,CURVA,EPSE,FNC,AMUC,AMVC)

C   DETERMINA A PARTIR DE UMA POSICAO ASSUMIDA P/ A LINHA NEUTRA A
C   NORMAL E OS MOMENTOS ULTIMOS DEVIDOS AO CONCRETO EM GUV

COMMON /R2/ FCK,GAMAC,IACO,ZCLAS,ES,FYK,GAMAS,DTETA,TETA1,BETA1,
1  TOLE,FNORM,NBT,NFT,NFTD,AREAC,FNMAX,AMXLI,AMyli,FNMIN,AMXMS,
2  AMYMS,FNMC,SGCD,FYD,EYDA,EYDB,EPD,AKEY,DBETA,AMOMX,AMOMY,
3  DFS,CURVA1
COMMON /R8/ GMCD,GMSD

      FNC=0.
      AMUC=0.
      AMVC=0.
      CALL SEPAC2(BETA,FNC,AMUC,AMVC,CURVA)

      RETURN
      END

SUBROUTINE LIMIT

C   CALCULA A MAXIMA E A MINIMA NORMAIS ULTIMAS RESISTIDAS PELA SECAO,
C   COM OS RESPECTIVOS MOMENTOS FLETORES ULTIMOS COMCOMITANTES.

COMMON /R1/ NSEC,NV,NA,NP,NCIRC(20),NEXT(100),X(10),Y(10),
INC1(2,10),BITO(2),CORDX(10),CORDY(10),IP,JP,BETAS(9),AREF,
CURVAS(9)
COMMON /R2/ FCK,GAMAC,IACO,ZCLAS,ES,FYK,GAMAS,DTETA,TETA1,BETA1,
1  TOLE,FNORM,NBT,NFT,NFTD,AREAC,FNMAX,AMXLI,AMyli,FNMIN,AMXMS,
2  AMYMS,FNMC,SGCD,FYD,EYDA,EYDB,EPD,AKEY,DBETA,AMOMX,AMOMY,
3  DFS,CURVA1
COMMON /R8/ GMCD,GMSD
COMMON /R6/ ADM

      ITT=0
      FYD=FYK/GAMAS
      EYDA=FYD/ES
      EYDB=EYDA+.002
      EPD= 7*EYDA
      NFTD=0
      WRITE(JP,100)

C   1) CALCULO DA NORMAL ULTIMA MAXIMA (COMPRESSAO UNIFORME)
C   1.1) CONTRIBUICAO DO CONCRETO
      SGCD = 0.85*FCK/GAMAC

      FNMC = AREAC*SGCD
      AMXLI=0.
      AMYLI=0.

C   1.2) CONTRIBUICAO DO ACO ACRESCENTADA
      FNMAX=FNMC
      SGSD=SIGMA(.002)
      DO 10 I=1,NBT

```



```

        FAUX=BITO(1)*SGSD
        FNMAX=FNMAX+FAUX
        AMXLI=AMXLI-FAUX*CORDY(I)
        AMYLI=AMYLI+FAUX*CORDX(I)
10    CONTINUE

C      II) CALCULO DA NORMAL ULTIMA MINIMA (TRACAO UNIFORME)
C      APENAS CONTRIBUICAO DO ACO

        SGSD=SIGMA(-0.01)
        FNMIN=0.
        AMXMS=0.
        AMYMS=0.
        DO 20 I=1,NBT
            FAUX=BITO(1)*SGSD
            FNMIN=FNMIN+FAUX
            AMXMS=AMXMS-FAUX*CORDY(I)
            AMYMS=AMYMS+FAUX*CORDX(I)
20    CONTINUE

        RETURN
        END

SUBROUTINE LINEUI(CURVA,BETA,VLNF)
C      DETERMINA A ORDENADA VLNF DA LINHA NEUTRA FICTICIA

        COMMON /R3/ U(100),V(100),CORDU(200),CORDV(200),VMAX,VMIN,VSMIN,H,
        ID,TETAS(9),FW(10)

        VLNF= VMAX-.8*(BETA*D)

        RETURN
        END

SUBROUTINE RES1
C      CALCULA AS CARACTERISTICAS GEOMETRICAS DA SECCAO FORNECIDA
C      TRANSLADA AS COORDENADAS DOS VERTICES E DAS BARRAS DE OXY PARA GX
C      FORNECE UM RESUMO QUE INCLUI CARACTERISTICAS DOS MATERIAIS,
C      CARACTERISTICAS GEOMETRICAS DA SECCAO FORNECIDA E UMA DESCRICAO
C      DOS ELEMENTOS CONSTANTES DAS TABELAS.

        DIMENSION XAUX(100),YAUX(100)
        COMMON /R1/ NSEC,NV,NA,NP,NCIRC(20),NEXT(100),X(10),Y(10),
        INCI(2,10),BITO(2),CORDX(10),CORDY(10),IP,JP,BETAS(9),AREF,
        CURVAS(9)
        COMMON /R2/ FCK,GAMAC,IACO,ZCLAS,ES,FYK,GAMAS,DTETA,TETA1,BETA1,
        TOLE,FNORM,NBT,NFT,NFTD,AREAC,FNMAX,AMXLI,AMYLI,FNMIN,AMXMS,
        AMYMS,FNMC,SGCD,FYD,EYDA,EYDB,EPD,AKEY,DBETA,AMOMX,AMOMY,
        DFS,CURVA1
        COMMON /R8/ GMCD,GMSD
        COMMON /R5/ XCG,YCG
        DATA BRANC/ ' '

        AREF=0.
        PI4=.7853981635
        AREAC=0.
        PERIM=0.
        AMXX=0.
        AMYY=0.
        DO 40 J=1,NP
            KK=NCIRC(J)
            DO 20 K=1,KK
                II=INCI(J,K)
                JJJ=IPESQ(NEXT,NV,II)
                XAUX(K)=X(JJJ)
                YAUX(K)=Y(JJJ)
20        CONTINUE
            CALL CIRC(XAUX,YAUX,KK,A,AMX,AMY)
            AREAC=AREAC+A
            AMXX=AMXX+AMX
            AMYY=AMYY+AMY
            DO 30 L=1,KK
                LL=L+1
                PI=DIST(XAUX(LL),YAUX(LL),XAUX(L),YAUX(L))
                PERIM=PERIM+PI
30        CONTINUE
40    CONTINUE
        XCG = AMYY/AREAC
        YCG = AMXX/AREAC
        DO 42 I=1,NV
            X(I) = X(I)-XCG
            Y(I) = Y(I)-YCG
42    CONTINUE
        DO 10 I=1,NBT
            CORDX(I)=CORDX(I)-XCG
            CORDY(I)=CORDY(I)-YCG
10    CONTINUE

```

```
AREF=0.04*AREAC
BITO(I)=AREF/REAL(NBT)
```

```
RETURN
```

```
STOP
END
```

```
SUBROUTINE RODA(TETA,COST,SENT)
```

```
C MONTA OS VETORES DE COORDENADAS U E V, CORDU E CORDV, RESPECTIVA
C DOS VERTICES E DAS BARRAS, NO SISTEMA GUV, OBTIDO DE OXY POR
C TRANSLACAO E ROTACAO TETA.
C DETERMINA VMAX, VMIN, VSMIN, H, D, COST E SENT.
```

```
COMMON /R1/ NSEC,NV,NA,NP,NCIRC(20),NEXT(100),X(10),Y(10),
* INCI(2,10),BITO(2),CORDX(10),CORDY(10),IP,JP,BETAS(9),AREF,
* CURVAS(9)
COMMON /R2/ FCK,GAMAC,IACO,ZCLAS,ES,FYK,GAMAS,DTETA,TETA1,BETA1,
1 TOLE,FNORM,NBT,NFT,NFTD,AREAC,FNMAX,AMXLI,AMYLI,FNMIN,AMXMS,
2 AMYMS,FNMC,SGCD,FYD,EYDA,EYDB,EPD,AKEY,DBETA,AMOMX,AMOMY,
3 DFS,CURVA1
COMMON /R8/ GMCD,GMSD
COMMON /R3/ U(100),V(100),CORDU(200),CORDV(200),VMAX,VMIN,VSMIN,H,
ID,TETAS(9),FW(10)
CONST=3.141592654/180.
COST = COS(TETA*CONST)
SENT = SIN(TETA*CONST)
VMAX = Y(1) * COST - X(1) * SENT
VMIN = VMAX
VSMIN= CORDY(1) * COST - CORDX(1) * SENT
DO 20 I = 1,NV
    U(I) = X(I)*COST+Y(I)*SENT
    V(I) = Y(I)*COST-X(I)*SENT
    IF(V(I).LE.VMAX) GO TO 10
    VMAX = V(I)
    GO TO 20
10    IF(V(I).GE.VMIN) GO TO 20
    VMIN = V(I)
20    CONTINUE
DO 30 I=1,NBT
    CORDU(I) = CORDX(I)*COST+CORDY(I)*SENT
    CORDV(I) = CORDY(I)*COST-CORDX(I)*SENT
    IF(CORDV(I).GE.VSMIN) GO TO 30
    VSMIN= CORDV(I)
30    CONTINUE
H=VMAX-VMIN
D=VMAX-VSMIN
RETURN
END
```

```
SUBROUTINE VERIF(FNF,FNI,F,IC)
```

```
C VERIFICA SE UMA FORCA NORMAL CAI DENTRO DO INTERVALO DE TOLERANCIA
C DA NORMAL FORNECIDA.
```

```
IC=0
IF(F.LT.FNI) IC=1
IF(F.GT.FNF) IC=2
RETURN
END
```

```
FUNCTION DIST(XI,YI,XF,YF)
```

```
C CALCULA A DISTANCIA ENTRE DOIS PONTOS.
```

```
DX=XI-XF
DY=YI-YF
X=DX*DX
Y=DY*DY
DIST=SQRT(X+Y)
RETURN
END
```

```
FUNCTION IPESQ(NEXT,N,K)
```

```
C PESQUISA EM NEXT (N POSICOES) O VALOR K.
```

```
DIMENSION NEXT(1)
DO 10 I=1,N
    IF(NEXT(I).EQ.K) GO TO 20
10    CONTINUE
IPESQ=0
RETURN
20    IPESQ=I
```

RETURN
END

FUNCTION SIGMA(EPS)

```
C   CALCULA A TENSAO SIGMA DO ACO A PARTIR DA DEFORMACAO EPS
C   UTILIZANDO O DIAGRAMA TENSAO-DEFORMACAO DE CALCULO DO ACO
      COMMON /R1/ NSEC,NV,NA,NP,NCIRC(20),NEXT(100),X(10),Y(10),
      INCI(2,10),BITO(2),CORDX(10),CORDY(10),IP,JP,BETAS(9),AREF,
      CURVAS(9)
      COMMON /R2/ FCK,GAMAC,IACO,ZCLAS,ES,FYK,GAMAS,DTETA,TETA1,BETA1,
1     TOLE,FNORM,NBT,NFT,NFTD,AREAC,FNMAX,AMXLI,AMYL1,FNMIN,AMXMS,
2     AMYMS,FNMC,SGCD,FYD,EYDA,EYDB,EPD,AKEY,DBETA,AMOMX,AMOMY,
3     DFS,CURVA1
      COMMON /R8/ GMCD,GMSD
      COMMON /R6/ ADM
      DATA A/'A',B/'B'/
      AEPS=ABS(EPS)
      IF(ZCLAS.NE.A) GO TO 100

C   I) ACO CLASSE A

      SIGMA=FYD
      IF(AEPS.LT.EYDA) SIGMA=ES*AEPS
      IF(EPS.LT.0.) SIGMA=-SIGMA
      RETURN

100  IF(ZCLAS.NE.B) GO TO 200

C   II) ACO CLASSE B

      IF(AEPS.LE.EPD) GO TO 120
      IF(AEPS.LE.EYDB) GO TO 140
      SIGMA=FYD
      GO TO 160
120  SIGMA=ES*AEPS
      GO TO 160
140  CTE=ES/FYD
      SIGMA=(7+1./(2.*CTE))*(SQRT(2025.+18.*CTE*(10.*CTE*AEPS-7.))-45.)
      *FYD
160  IF(EPS.LT.0.) SIGMA=-SIGMA
      RETURN

      STOP
      END
```

FUNCTION UINT(UI,VI,UF,VF,VLNF)

```
C   CALCULA ABSCISSA UINT DA INTERSECCAO DO SEGMENTO(UI,VI)-(UF,VF)
C   COM A LINHA NEUTRA FICTICIA (LNF).

      UINT=UI+(VLNF-VI)*(UF-UI)/(VF-VI)
      RETURN
      END
```

SUBROUTINE SEPAC2(BETA,FNC,AMUC,AMVC,CURVA)

```
C   CALCULA OS ESFORCOS SOLICITANTES ULTIMOS, FNC, AMUC E AMVC,
C   RESISTIDOS PELO CONCRETO

      COMMON /R1/ NSEC,NV,NA,NP,NCIRC(20),NEXT(100),X(10),Y(10),
      INCI(2,10),BITO(2),CORDX(10),CORDY(10),IP,JP,BETAS(9),AREF,
      CURVAS(9)
      COMMON /R2/ FCK,GAMAC,IACO,ZCLAS,ES,FYK,GAMAS,DTETA,TETA1,BETA1,
1     TOLE,FNORM,NBT,NFT,NFTD,AREAC,FNMAX,AMXLI,AMYL1,FNMIN,AMXMS,
2     AMYMS,FNMC,SGCD,FYD,EYDA,EYDB,EPD,AKEY,DBETA,AMOMX,AMOMY,
3     DFS,CURVA1
      COMMON /R8/ GMCD,GMSD
      COMMON /R3/ U(100),V(100),CORDU(200),CORDV(200),VMAX,VMIN,VSMIN,H,
      ID,TETAS(9),FW(10)

      VLN=VMAX-(BETA*D)
      V2=(0.002+CURVA*(VMAX-BETA*D))/CURVA
      FNC=0.
      AMUC=0.
      AMVC=0.
      DO 50 I=1,NP
          NN=NCIRC(I)+1
          INCI(I,NN)=INCI(I,1)
          IK=NCIRC(I)
          DO 40 J=1,IK
              VJ=V(INCI(I,J))
              VJ1=V(INCI(I,J+1))
              UJ=U(INCI(I,J))
```

```

UJ1=U(INCI(I,J+1))
IF (VJ.LT.VLN.AND.VJ.NE.VJ1) THEN
  IF (VJ1.LE.VLN) GOTO 40
  IF (VJ1.LE.V2) THEN
    ULN=COORD(UJ,UJ1,VJ,VJ1,VLN)
    CALL PARA(ULN,UJ1,VLN,VJ1,V2,VLN,FNC,AMUC,AMVC)
    GOTO 40
  ENDIF
  IF (VJ1.GT.V2) THEN
    ULN=COORD(UJ,UJ1,VJ,VJ1,VLN)
    U2=COORD(UJ,UJ1,VJ,VJ1,V2)
    CALL PARA(ULN,U2,VLN,V2,V2,VLN,FNC,AMUC,AMVC)
    CALL CONST(U2,UJ1,V2,VJ1,FNC,AMUC,AMVC)
    GOTO 40
  ENDIF
ELSEIF (VJ.LT.V2.AND.VJ.NE.VJ1) THEN
  IF (VJ1.LE.VLN) THEN
    ULN=COORD(UJ,UJ1,VJ,VJ1,VLN)
    CALL PARA(UJ,ULN,VJ,VLN,V2,VLN,FNC,AMUC,AMVC)
    GOTO 40
  ENDIF
  IF (VJ1.LE.V2) THEN
    CALL PARA(UJ,UJ1,VJ,VJ1,V2,VLN,FNC,AMUC,AMVC)
    GOTO 40
  ENDIF
  IF (VJ1.GT.V2) THEN
    U2=COORD(UJ,UJ1,VJ,VJ1,V2)
    CALL PARA(UJ,U2,VJ,V2,V2,VLN,FNC,AMUC,AMVC)
    CALL CONST(U2,UJ1,V2,VJ1,FNC,AMUC,AMVC)
    GOTO 40
  ENDIF
ELSEIF (VJ.GE.V2.AND.VJ.NE.VJ1) THEN
  IF (VJ1.LT.VLN) THEN
    ULN=COORD(UJ,UJ1,VJ,VJ1,VLN)
    U2=COORD(UJ,UJ1,VJ,VJ1,V2)
    CALL PARA(U2,ULN,V2,VLN,V2,VLN,FNC,AMUC,AMVC)
    CALL CONST(UJ,U2,VJ,V2,FNC,AMUC,AMVC)
    GOTO 40
  ENDIF
  IF (VJ1.LT.V2) THEN
    U2=COORD(UJ,UJ1,VJ,VJ1,V2)
    CALL PARA(U2,UJ1,V2,VJ1,V2,VLN,FNC,AMUC,AMVC)
    CALL CONST(UJ,U2,VJ,V2,FNC,AMUC,AMVC)
    GOTO 40
  ENDIF
  IF (VJ1.GE.V2) THEN
    CALL CONST(UJ,UJ1,VJ,VJ1,FNC,AMUC,AMVC)
    GOTO 40
  ENDIF
ENDIF
40 CONTINUE
50 CONTINUE
RETURN
END

FUNCTION COORD(U1,U2,V1,V2,V)
C DETERMINA A ABSCISSA CORRESPONDENTE A ORDENADA V
COORD=U1+(U2-U1)*(V-V1)/(V2-V1)
RETURN
END

SUBROUTINE CONST(UK,UK1,VK,VK1,FNC,AMUC,AMVC)
C CALCULA OS ESFORÇOS SOLICITANTES ÚLTIMOS, FNC, AMUC E AMVC,
C RESISTIDOS PELO CONCRETO, QUANDO O SEGMENTO SE ENCONTRA NA REGIÃO
C DE TENSÃO CONSTANTE.
COMMON /R1/ NSEC,NV,NA,NP,NCIRC(20),NEXT(100),X(10),Y(10),
INCI(2,10),BITO(2),CORDX(10),CORDY(10),IP,JP,BETAS(9),AREF,
CURVAS(9)
COMMON /R2/ FCK,GAMAC,IACO,ZCLAS,ES,FYK,GAMAS,DTETA,TETA1,BETA1,
1 TOLE,FNORM,NBT,NFT,NFTD,AREAC,FNMAX,AMXLI,AMYLI,FNMIN,AMXMS,

```

```

2  AMYMS,FNMC,SGCD,FYD,EYDA,EYDB,EPD,AKEY,DBETA,AMOMX,AMOMY,
3  DFS,CURVAI
COMMON /R8/ GMCD,GMSD
COMMON /R3/ U(100),V(100),CORDU(200),CORDV(200),VMAX,VMIN,VSMIN,H,
ID,TETAS(9),FW(10)

FNC=FNC+(SGCD/2.)*(VK1-VK)*(UK+UK1)
AMUC=AMUC-(SGCD/6.)*(VK1-VK)*(2.*UK*VK+UK*VK1+UK1*VK+2.*UK1*VK1)
AMVC=AMVC+(SGCD/6.)*(VK1-VK)*((UK+UK1)**2-UK*UK1)

RETURN
END

SUBROUTINE PARA(UK,UK1,VK,VK1,V2,VLN,FNC,AMUC,AMVC)

C  CALCULA OS ESFORÇOS SOLICITANTES ULTIMOS, FNC, AMUC E AMVC.
C  RESISTIDOS PELO CONCRETO,QUANDO O SEGMENTO SE ENCONTRA NA REGIAO
C  DE COM DISTRIBUICAO PARABOLICA DE TENSAO.

COMMON /R1/ NSEC,NV,NA,NP,NCIRC(20),NEXT(100),X(10),Y(10),
* INCI(2,10),BITO(2),CORDX(10),CORDY(10),IP,JP,BETAS(9),AREF,
* CURVAS(9)
COMMON /R2/ FCK,GAMAC,IACO,ZCLAS,ES,FYK,GAMAS,DTETA,TETA1,BETA1,
1  TOLE,FNORM,NBT,NFT,NFTD,AREAC,FNMAX,AMXLI,AMYLI,FNMIN,AMXMS,
2  AMYMS,FNMC,SGCD,FYD,EYDA,EYDB,EPD,AKEY,DBETA,AMOMX,AMOMY,
3  DFS,CURVAI
COMMON /R8/ GMCD,GMSD
COMMON /R3/ U(100),V(100),CORDU(200),CORDV(200),VMAX,VMIN,VSMIN,H,
ID,TETAS(9),FW(10)

A=-SGCD/(V2-VLN)**2
A1=A*(VK1-VK)**2
B1=2.*(VK-V2)*(VK1-VK)*A
C1=(VK-VLN)*(VK+VLN-2.*V2)*A

* FNC=FNC+(VK1-VK)/12.*(UK1*(3.*A1+4.*B1+6.*C1)+
UK*(A1+2.*B1+6.*C1))

* AMUC=AMUC-(VK1-VK)/60.*(UK*VK*(2.*A1+5.*B1+20.*C1)+(UK*VK1+
UK1*VK)*(3.*A1+5.*B1+10.*C1)+UK1*VK1*(12.*A1+15.*B1+20.*C1))

* AMVC=AMVC+(VK1-VK)/120.*(UK**2*(2.*A1+5.*B1+20.*C1)+
2.*UK*UK1*(3.*A1+5.*B1+10.*C1)+UK1**2*(12.*A1+15.*B1+20.*C1))

RETURN
END

```