

**UNIVERSIDADE FEDERAL DE SÃO CARLOS**  
**CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ESTRUTURA E CONSTRUÇÃO CIVIL**

**PROGRAMA COMPUTACIONAL PARA ANÁLISE E VERIFICAÇÃO DE SEÇÕES  
DE CONCRETO ARMADO E PROTENDIDO COM FLEXÃO OBLÍQUA  
COMPOSTA**

**LEONARDO MARTINS E SILVA**

**São Carlos**  
**2015**

**UNIVERSIDADE FEDERAL DE SÃO CARLOS**  
**CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ESTRUTURAS E CONSTRUÇÃO CIVIL**

**PROGRAMA COMPUTACIONAL PARA ANÁLISE E VERIFICAÇÃO DE SEÇÕES  
DE CONCRETO ARMADO E PROTENDIDO COM FLEXÃO OBLÍQUA  
COMPOSTA**

**LEONARDO MARTINS E SILVA**

Dissertação apresentada ao Programa de Pós Graduação em Estruturas e Construção Civil da Universidade Federal de São Carlos, como parte dos requisitos para obtenção do título de Mestre em Estruturas e Construção Civil.

**Área de Concentração:** Análise Estrutural

**Orientador:** Prof. Dr. Roberto Chust Carvalho

**São Carlos**  
**2015**

**Ficha catalográfica elaborada pelo DePT da  
Biblioteca Comunitária da UFSCar**

S586pc

Silva, Leonardo Martins e.

Programa computacional para análise e verificação de seções de concreto armado e protendido com flexão oblíqua composta / Leonardo Martins e Silva. -- São Carlos : UFSCar, 2015.

189 f.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2015.

1. Concreto. 2. Programas de computador. 3. Flexão (Engenharia civil). 4. Concreto armado. 5. Concreto protendido. I. Título.

CDD: 624.1834 (20<sup>a</sup>)



# UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia  
Programa de Pós-Graduação em Estruturas e Construção Civil

---

## Folha de Aprovação

---

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado do candidato Leonardo Martins e Silva, realizada em 31/03/2015:

---

Prof. Dr. Roberto Chust Carvalho  
UFSCar

---

Prof. Dr. Fernando Menezes de Almeida Filho  
UFSCar

---

Prof. Dr. Marcos Antonio Marino  
UFPR

## RESUMO

No presente trabalho, apresenta-se um programa livre e gratuito de análises e verificações de estados-limite último e de serviço de seções de concreto armado e protendido com forma poligonal submetidas à flexão composta oblíqua. Além disso, o programa calcula também a relação momento-normal-curvatura ( $M$ ,  $N$ ,  $1/r$ ) das seções, que pode ser usada em outras análises, como por exemplo, o método geral para o cálculo de pilares. Para isso se usa a programação orientada ao objeto, a fim de que o presente trabalho possa ser retomado e alterado, dando subsídio técnico para trabalhos futuros da mesma linha, por esse motivo fornece-se as rotinas de cálculo juntamente ao programa. As linhas e rotinas de cálculo constam com explicações de seu funcionamento para incentivar seu uso posterior como parte de outros programas. A entrada dos dados é feita de forma de formulário e o programa permite a troca de dados, tanto de entrada como de saída através de arquivos do tipo texto. Os critérios adotados seguem a norma brasileira ABNT NBR 6118:2014, que em relação à versão anterior apresenta sugestões para concretos com resistência superior a 50 MPa. A estrutura das rotinas é apresentada de maneira que critérios de outras normas, ou ainda atualizações da mesma norma brasileira referenciada, possam ser facilmente implementadas. Apresenta-se vários exemplos de validação do programa em diversas situações e finalmente comentadas e listadas as ampliações que podem ser feitas.

Palavras-chave: Flexão Oblíqua Composta; Relação momento-curvatura; Concreto Armado, Concreto Protendido; programas computacionais.

## ABSTRACT

In this paper, a free analysis program is presented for the verification of ultimate and serviceability limit states of reinforced or prestressed concrete sections with polygonal shape submitted to axial forces and biaxial bending. In addition, the program also calculates the bending-axial force-curvature relation ( $M, N, 1 / r$ ) of the sections, which can be used in further analysis, for example, the general method for calculating pillars. For this is used the object-oriented programming, so that this work can be resumed and amended, providing technical material for future work in the same line, therefore provides that the calculation routines along with the program. The programming lines and routines includes explanations of its operation to encourage its further use as part of other programs. The data entry is presented as a form and the program enables the exchange of data, both input and output via text type files. The criteria adopted follows the brazilian standard ABNT NBR 6118: 2014, which compared to the previous version features suggestions for concrete with superior characteristic strength higher than 50 MPa. The structure of the routines is presented so that criteria of other standards, or updates of the same Brazilian standard referenced, can be easily implemented. Some validation examples for the program are presented in different situations and finally commented and listed some other extensions which can be made.

**Keywords:** Biaxial Bending with Axial Force; Bending Moment - Curvature Relation; Reinforced Concrete; Prestressed Concrete; Computer program.

## LISTA DE FIGURAS

Figura 1 - Estádios de deformação	19
Figura 2 – Convenção usada para os momentos na seção.	22
Figura 3 - Domínios de deformação em ELU para $f_{ck} \leq 50$ MPa.	23
Figura 4 - Domínios de deformação em ELU para $f_{ck} > 50$ MPa.	23
Figura 5 - As três regiões de ruptura.	24
Figura 6 - Deformações na região I.	25
Figura 7 - Deformações na região II.	25
Figura 8 - Deformações na região III.	26
Figura 9 - Diagrama tensão-deformação do concreto.	27
Figura 10 - Diagrama tensão-deformação do aço.	28
Figura 11 – Relação tensão-deformação da armadura ativa	29
Figura 12 - Superfícies de interação.	32
Figura 13 - Seção genérica sob flexão composta oblíqua – coordenadas iniciais de referência $xy$ .	33
Figura 14 – Exemplo de conformação de curvatura na seção e convenção indesejada para $\alpha$	36
Figura 15 – Exemplo de conformação de curvatura na seção e convenção usada para $\alpha$	37
Figura 16 – Relações momento-curvatura real e supondo estágio II puro.	38
Figura 17 – Relação momento-curvatura: aspecto segundo a norma	38
Figura 18 – Diagrama tensão-deformação do concreto à tração.	41
Figura 19 – Exemplo de figura plana definida por uma poligonal.	44
Figura 20 – Translação de coordenadas em seção genérica.	45
Figura 21 - Faixa infinitesimal $dy$ na área de concreto.	47
Figura 22 – Área de concreto sujeita a equações de tensão diferentes.	49
Figura 23 - Variáveis referentes às tensões no concreto.	50
Figura 24 – Obtenção do valor de $x_{\alpha}$ e $x_{ec2}$	50
Figura 25 - Relação entre posição vertical e deformação das fibras de concreto.	51
Figura 26 - Exemplo de situação de cálculo.	58
Figura 27 - Algoritmos para resolução da FCO.	61
Figura 28 – Algoritmo usado para obtenção do momento-curvatura.	63
Figura 29 – Algoritmo para verificação de estado-limite de serviço.	66
Figura 30 – Interpretação Geométrica do Método de Newton	68

Figura 31 – Interpretação Geométrica do Método da Secante	69
Figura 32 – Ponto de deformação $\varepsilon_c$ .	75
Figura 33 – Programa elaborado, OblqCALCO, com os dados de análise.	82
Figura 34 - Resultados do exemplo 1, segundo o programa elaborado.	83
Figura 35 – Armadura do Exemplo 1, segundo programa de cálculo de armadura.	83
Figura 36 – Seção a ser dimensionada no exemplo 2.	84
Figura 37 – Ábaco usado para resolução do exemplo 2.	85
Figura 38 – Altura da armadura passiva dimensionada da seção do exemplo 2.	86
Figura 39 – Resultados do exemplo 2 pelo OblqCALCO.	86
Figura 40 - Seção a ser dimensionada no exemplo 3.	87
Figura 41 – Ábaco relativo ao exemplo 3.	88
Figura 42 – Seção dimensionada no exemplo 3 e posições da armadura.	89
Figura 43 – Seção aplicada no programa, seus resultados e um zoom no mesmo gráfico.	89
Figura 44 – Medidas da seção do exemplo 4.	90
Figura 45 – Diagrama de momentos resistentes para $N_d = 1000\text{kN}$ (compressão)	91
Figura 46 – Diagrama de momentos resistentes para $N_d = -500\text{ kN}$ (Tração).	92
Figura 47 – Seção a ser analisada no exemplo 5, com cotas em cm.	93
Figura 48 – Resultados segundo este trabalho à esquerda e do Oblíqua à direita.	93
Figura 49 – Seção do exemplo 6.	94
Figura 50 – Relações Momento-Curvatura sendo a força normal característica 200 kN.	95
Figura 51 - Relações Momento-Curvatura sendo a força normal característica 420 kN	95
Figura 52 - Seção Transversal do Exemplo 4.	96
Figura 53 – Gráfico Momento – Curvatura adimensional para $\nu = 0,0$ .	98
Figura 54 – Gráfico Momento – Curvatura adimensional para $\nu = 0,2$ .	99
Figura 55 – Gráfico Momento – Curvatura adimensional para $\nu = 0,5$ .	99
Figura 56 – Poligonal do exemplo 8.	100
Figura 57 – Seção referente ao exemplo 9	102
Figura 58 – Seção do exemplo 9 homogeneizada	103
Figura 59 – Dados geométricos da seção do exemplo 5.	104
Figura 60 – Resultados do exemplo 10	106
Figura 61 – Parte dos resultados do exemplo 11.	107
Figura 62 – Seção estudada no exemplo 12.	108
Figura 63 – Resultados da verificação do exemplo 12.	111
Figura 64 – Resultados do programa sendo copiados para uma planilha.	119

Figura 65 – Resultados do programa aplicados e modificados numa planilha.	119
Figura 66 – Cálculo da força no concreto por integração tripla.	125
Figura A. 1 – Ambiente de trabalho do programa OblqCALCO	132
Figura A. 2 – Ambiente de configuração dos materiais no programa OblqCALCO	133
Figura A. 3 – Seção exemplo de entrada no programa OblqCALCO	134
Figura A. 4 – Coordenadas dos pontos da seção exemplo.	135
Figura A. 5 – Resultado gráfico da análise de estado-limite último da seção exemplo.	136
Figura A. 6 – Resultado da análise de estado-limite último da seção exemplo como texto.	137
Figura A. 7 – Envoltória de momentos resistentes da seção exemplo.	137
Figura A. 8 – Resultado gráfico da análise momento-curvatura para a seção exemplo.	138
Figura A. 9 – Variáveis utilizadas no ambiente “Calcular Solicitações”.	139
Figura A. 10 – Entrada de dados da armadura ativa.	141

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>10</b>
1.1	JUSTIFICATIVA .....	12
1.2	OBJETIVOS .....	14
1.3	METODOLOGIA DE PESQUISA .....	14
1.4	ESTRUTURA DO TRABALHO .....	15
<b>2</b>	<b>REVISÃO BIBLIOGRAFICA E CONCEITOS FUNDAMENTAIS.....</b>	<b>17</b>
2.1	ALGUNS TRABALHOS .....	17
2.2	CONSIDERAÇÕES INICIAIS .....	18
<b>2.2.1</b>	<b>Convenção de sinais .....</b>	<b>21</b>
2.3	DOMÍNIOS DE DEFORMAÇÃO .....	22
2.4	RELAÇÕES TENSÃO-DEFORMAÇÃO .....	26
2.5	VERIFICAÇÃO ESTADO-LIMITE ÚLTIMO .....	30
2.6	COMPORTAMENTO E VERIFICAÇÕES EM SERVIÇO.....	33
<b>2.6.1</b>	<b>Gráfico Momento-Normal-Curvatura.....</b>	<b>35</b>
<b>2.6.2</b>	<b>Verificações de estado limite de abertura de fissuras.....</b>	<b>39</b>
<b>2.6.3</b>	<b>Verificações de estado limite de formação de fissuras .....</b>	<b>39</b>
<b>2.6.4</b>	<b>Verificações de estado limite de descompressão .....</b>	<b>41</b>
<b>3</b>	<b>ROTINAS DE CÁLCULO .....</b>	<b>42</b>
3.1	PREPARAÇÃO DOS DADOS .....	42
<b>3.1.1</b>	<b>Características geométricas .....</b>	<b>43</b>
<b>3.1.2</b>	<b>Translação da origem do eixo de coordenadas .....</b>	<b>45</b>
<b>3.1.3</b>	<b>Rotação da origem do eixo de coordenadas .....</b>	<b>45</b>
3.2	CÁLCULO DA FORÇA E MOMENTOS DO CONCRETO .....	46
3.3	CÁLCULO DA FORÇA E MOMENTOS DA ARMADURA PASSIVA .....	58
3.4	CÁLCULO DA FORÇA E MOMENTOS DA ARMADURA ATIVA .....	59
3.5	CÁLCULOS FINAIS DA ETAPA.....	60
3.6	ESTADO-LIMITE ÚLTIMO .....	61
3.7	ESTADO-LIMITE DE SERVIÇO .....	62
<b>3.7.1</b>	<b>Gráfico Momento-Curvatura .....</b>	<b>62</b>
<b>3.7.2</b>	<b>Verificações dos estados-limite de serviço .....</b>	<b>65</b>
3.8	MÉTODO DE NEWTON E MÉTODO DA SECANTE .....	67
<b>3.8.1</b>	<b>Estado-limite último .....</b>	<b>71</b>

3.8.2	<b>Gráfico Momento-Normal-Curvatura.....</b>	<b>73</b>
3.8.3	<b>Verificação de estados-limite de fissuração e descompressão.....</b>	<b>74</b>
<b>4</b>	<b>VALIDAÇÃO DOS RESULTADOS .....</b>	<b>80</b>
4.1	EXEMPLO 1 – FLEXÃO SIMPLES .....	81
4.2	EXEMPLO 2 – FLEXÃO COMPOSTA OBLÍQUA.....	84
4.3	EXEMPLO 3 – FLEXÃO COMPOSTA OBLÍQUA.....	87
4.4	EXEMPLO 4 – DIAGRAMA DE MOMENTOS RESISTENTES .....	89
4.5	EXEMPLO 5 – DIAGRAMA DE MOMENTOS RESISTENTES COM OBLIQUA 1.0. ....	92
4.6	EXEMPLO 6 – RELAÇÃO MOMENTO X CURVATURA USANDO PCALC .....	94
4.7	EXEMPLO 7 – GRÁFICO MOMENTO X NORMAL X CURVATURA COM RIBEIRO (2011).....	96
4.8	EXEMPLO 8 – VERIFICAÇÕES DE SERVIÇO .....	99
4.9	EXEMPLO 9 – VERIFICAÇÃO DE SERVIÇO DE ABERTURA DE FISSURAS COM PROTENSÃO.....	101
4.10	EXEMPLO 10 – FLEXÃO RETA EM CONCRETO C90.....	105
4.11	EXEMPLO 11 – ESTADO-LIMITE ÚLTIMO DE ARMADURA ATIVA.....	107
4.12	EXEMPLO 12 – ESTADO-LIMITE DE ABERTURA DE FISSURAS.....	108
<b>5</b>	<b>APRESENTAÇÃO DO USO DAS ROTINAS E DO PROGRAMA.....</b>	<b>112</b>
5.1	ROTINAS;.....	112
5.1.1	Cálculo das características geométricas da seção;.....	113
5.1.2	Translação do eixo de coordenadas da seção;.....	113
5.1.3	Rotação do eixo de coordenadas da seção;.....	114
5.1.4	Cálculo das forças normais da seção de concreto;.....	114
5.1.5	Cálculo dos momentos da seção de concreto;.....	115
5.1.6	Cálculo da tensão na armadura passiva;.....	115
5.1.7	Cálculo da tensão na armadura ativa;.....	116
5.2	RESUMO DAS FUNÇÕES APRESENTADAS E SEUS PARÂMETROS E VARIÁVEIS;.....	116
5.3	PROCEDIMENTOS PARA EXPORTAR RESULTADOS DO PROGRAMA .....	118
<b>6</b>	<b>CONSIDERAÇÕES FINAIS.....</b>	<b>120</b>
6.1	SUGESTÕES PARA TRABALHOS FUTUROS.....	123
<b>7</b>	<b>REFERÊNCIAS .....</b>	<b>126</b>
<b>A.</b>	<b>APÊNDICE 1 – MANUAL DO PROGRAMA OBLQCALCO .....</b>	<b>130</b>

A.1	CARACTERÍSTICAS GERAIS E ENTRADA DE DADOS.....	130
A.2	ENTRADA DE DADOS .....	134
A.3	EXPORTAÇÃO DE DADOS .....	135
A.4	ABA “ELU” .....	136
A.5	ABA “MOMENTO-CURVATURA” .....	138
A.6	ABA “CALCULAR SOLICITAÇÕES” .....	139
A.7	ABA “VERIFICAR SOLICITAÇÕES - ELS” .....	140
A.8	USO DE PROTENSÃO .....	141
<b>B.</b>	<b>APÊNDICE 2 – CÓDIGOS DO PROGRAMA OBLQCALCO .....</b>	<b>142</b>
B.1	ROTINA DE ESTADO-LIMITE ÚLTIMO .....	142
B.2	ROTINA DA RELAÇÃO MOMENTO-CURVATURA.....	149
B.3	ROTINA DA VERIFICAÇÃO DE ESTADO-LIMITE DE SERVIÇO .....	155
B.4	ROTINA PARA CÁLCULAR SOLICITAÇÕES .....	165
B.5	ROTINAS.....	168

## 1 INTRODUÇÃO

Santos (1994), no prefácio do livro “Sub-rotinas básicas do dimensionamento de concreto armado” salienta a importância do uso de computadores na engenharia de estruturas, como uma das justificativas de sua publicação: o advento da era eletrônica tornou as normas mais exigentes quanto à sofisticação de cálculo, tornando inexecutável o cálculo de concreto armado sem o uso do computador. O autor também destaca a constante presença de programas não abertos, isto é, cujo funcionamento é muitas vezes uma incógnita.

No que diz respeito à importância do processamento de dados, a aproximação de funções e suas integrais a cálculos e processos de aproximações numéricas aumenta expressivamente o número de operações necessárias para a resolução das mesmas. Quando se trata de análise estrutural, muitas vezes as operações e métodos seriam inviáveis de serem resolvidas sem recursos computacionais, isto é, usando apenas recursos humanos. Assim, para facilitar os cálculos, muitos métodos foram elaborados: por exemplo, para dimensionamento em flexão composta oblíqua, os ábacos de Venturini e Rodrigues (1987).

Com o tempo, alguns programas gratuitos (porém não livre) foram ganhando espaço, exemplos típicos são o Ftool, elaborado por Martha (2002) da PUC-Rio, e os programas Normal 1.3 (2001) e Oblíqua 1.0 (2001), elaborados pelos professores e alunos da UFPR (2001), que são programas de análise estrutural e verificações de seções de concreto armado à solicitações normais, respectivamente. Tratam-se de programas gratuitos disponíveis aos profissionais para auxiliar em suas atividades, tanto nas áreas de pesquisa e estudo, quanto no mercado de trabalho.

Porém, a capacidade dos computadores tem aumentado com o tempo, e atualmente é possível que sejam feitos cálculos mais numerosos em menos tempo que antes. Assim, vários processos numéricos que anteriormente seriam inviáveis devido ao número de operações, hoje podem ser executados.

Aliado a essa nova realidade da computação, tem-se o problema da dificuldade de determinação dos esforços resistentes e dimensionamento da armadura das seções submetidas à flexão composta oblíqua (F. C. O) ou Flexão oblíqua composta – (F. O. C.), cujos problemas são resolvidos impondo-se a eles certas restrições, como descreve Pinheiro, Baraldi e Porem (2009). Segundo o autor, são adotadas restrições para que seja possível resolver o problema, como a adoção da forma da seção, e distribuição das barras em relação ao tamanho da peça, e mesmo assim a solução com as equações e condições de compatibilidade é muito trabalhosa e requer uso de computadores.

Alguns programas já auxiliam neste problema: o Oblíqua 1.0 (2001) e o programa apresentado por Medeiros (2004), o nFocca. O Oblíqua trata-se de um programa gratuito para verificação de seções submetidas à flexão composta oblíqua em estado-limite último. Nele, pode-se verificar seções de diversas formas, e com diversas posições da armadura. O resultado de sua análise é o diagrama que, para uma força normal pré-estabelecida, apresenta os momentos resistentes em cada direção.

Já o nFocca é um programa livre, e por possuir código aberto, apresenta e detalha todos os seus procedimentos de cálculo, possibilitando que estes sejam estudados e utilizados. Portanto, além de fornecer uma ferramenta que possui verificações e dimensionamento, há também as rotinas e as linhas de programação internas do programa, que podem ser estudadas e modificadas. Medeiros (2004) também entra em detalhes sobre os processos de cálculo usados no programa, que aborda o problema da F.O.C. de maneira semelhante à Santos (1994). Inclusive, Medeiros (2004) aborda a tensão no concreto usando uma equação paralela à equação da norma, que era possível apenas quando o expoente  $n$  tinha valor fixo 2 (ver Figura 9).

Outro trabalho que acompanha a mesma linha é o de Smaniotto (2005), que apresenta detalhadamente o raciocínio proposto por Santos (1994) ao calcular os esforços resistentes do concreto, processo este denominado na bibliografia como “processo de integração de trapézios”. A leitura da dissertação citada é complementar quanto à aplicação deste método em relação ao apresentado por Santos (1994).

Com a atualização da norma de concreto, anteriormente ABNT NBR 6118:2007, hoje conhecida como ABNT NBR 6118:2014, uma das alterações principalmente no que diz respeito à cálculo de elementos é a introdução do cálculo de elementos com resistência característica a compressão ( $f_{ck}$ ) superior a 50MPa e inferior a 90MPa. Portanto, os programas citados, por ser de data anterior, ainda não incluem essas novas diretrizes da norma, devendo estas ser inclusas no programa elaborado.

E dando sequência ao cálculo, com uma rotina que calcula as resistências axiais e momentos resistentes em cada direção em função de deformações pré-estabelecidas pode-se fazer a verificação ao estado-limite de serviço das seções em flexão composta oblíqua pela formulação do diagrama momento-curvatura (M-N-1/r) das seções. O mesmo é usado em várias situações e segundo a ABNT NBR 6118:2014, trata-se de um método para consideração os efeitos da não-linearidade.

Além disso, a mesma norma apresenta novamente a utilização das relações momento-curvatura na utilização do método geral para a determinação dos efeitos locais de segunda

ordem (seção 15.8.3), sendo dentre os métodos citados, sendo o mais refinado para tal cálculo. Além disso, a norma também apresenta o método do pilar-padrão acoplado a diagramas M, N e  $1/r$  (curvatura), em que o cálculo da curvatura também pode ser utilizado.

Considerando-se também a importância e a vasta utilização da protensão, e seu efeito positivo nas construções em concreto, é viável que esta possa ser considerada: as peças protendidas podem ser sujeitas à flexão composta oblíqua, e é interessante que possam ser estudadas como tal. Para isso, é necessário que se verifiquem os estados-limites de serviço dessa seção, como transcreve a ABNT NBR 6118:2014.

## 1.1 JUSTIFICATIVA

Dentre as utilidades deste trabalho, deseja-se possibilitar a verificação de seções submetidas a esforços normais seguindo as diretrizes da ABNT NBR 6118:2014. Esta cita verificações de estados-limites últimos e de serviço, sendo que consta na abordagem do programa verificação dos estados-limite de serviço referentes à protensão: abertura de fissuras, formação de fissuras e descompressão.

Ainda seguindo as verificações da mesma norma, outra contribuição deste trabalho diz respeito à obtenção da relação momento-curvatura, relação esta que é imprescindível em várias análises, tendo como principal o método geral, obrigatória para pilares com esbeltez muito elevada, por exemplo.

Alguns trabalhos nessa linha, incluem Araújo (2009), que utiliza diagramas momento-curvatura para mostrar variação de ductilidade de peças solicitadas à flexão simples (vigas e lajes) quando dimensionadas sob diferentes alturas de linha neutra. Ceccon (2008), estuda o problema de flexão oblíqua composta dividindo-a em duas direções principais de inércia, considerando uma rigidez obtida no diagrama momento-curvatura, e compara o resultado dessa análise com a integração numérica com a curvatura obtida ponto a ponto (integrando pela Quadratura de Gauss). E Paula (1988) detalha o uso da curvatura na integração para o método geral, gerando um programa para flexão normal composta, inclusive.

Outra contribuição do programa diz respeito às considerações geométricas da seção, fissurada ou não: o diagrama M, N,  $1/r$  possibilita verificar a seção quanto ao estado-limite de serviço com versatilidade.

Além disso, acompanhando os avanços na área da informática, as normas podem se tornar mais exigentes quanto ao número e dificuldade das verificações estruturais, valorizando ferramentas de análise de estruturas. Ainda mais como um programa livre: dará a possibilidade de continuação e alteração do programa quanto a outras normas e critérios.

Entende-se aqui por programa livre aquele em são fornecidos o programa executável, a listagem aberta e a documentação pertinente. Tais programas que auxiliem nas verificações podem ser de grande utilidade, pois:

- por serem gratuitos não geram restrições financeiras de acesso;
- podem ser usados para ensino porque vêm acompanhados de informações dos procedimentos permitindo que o usuário possa aprender;
- permitem a constante atualização justamente por terem códigos e informações fornecidas;
- podem ser usados aplicados a outros programas como sub-rotina ou trechos encapsulados;
- por ser estruturado pode ser facilmente adaptado a qualquer tipo de norma ou material;

Para pesquisadores na linha de aplicação de sub-rotinas e algoritmos de cálculo em estruturas, tal trabalho pode servir de exemplo e aprendizado no que diz respeito à implementação de algoritmos e equações em forma de linguagem de programação, podendo auxiliar na elaboração de eventuais programas na mesma linha, dando sequência aos trabalhos já realizados.

A linguagem orientada ao objeto, que será a empregada neste trabalho, permitirá que cada variável ou análise seja tratada numa unidade isolada, facilitando o entendimento e modificação do programa. E por ser de código aberto, poderá ser aplicado a outros programas, economizando trabalho e fornecendo um código que pode ser reaplicado e atualizado. Podendo acompanhar outros programas e novas disposições de normas, ou mesmo outras normas que seguem diretrizes para dimensionamento e verificação diferentes da NBR 6118:2014, que será a norma focada neste trabalho.

O programa pode ser usado para verificação, dimensionamento e nas situações em serviço e estado-limite último. É muito comum em análises de concreto armado ser necessário calcular a rigidez de elementos (principalmente para situações em serviço), sendo importante neste caso o cálculo da curvatura na seção. Em outras situações o que importa são os esforços resistentes. Programando com orientação ao objeto pode-se conseguir os dois objetivos praticamente com os mesmos recursos, tornando assim o programa mais completo e extraindo mais recursos das mesmas ferramentas.

Uma observação importante de se fazer um programa com orientação ao objeto é que esta permitirá calcular a relação tensão-deformação dos materiais isoladamente das outras

rotinas: consideração esta importante devido às possíveis alterações e adições relativas a concretos de alta resistência.

## 1.2 OBJETIVOS

O objetivo geral desse trabalho é apresentar um programa de cálculo com rotinas de verificação de seções de concreto armado e protendido submetidas à flexão composta oblíqua e que calcule a curvatura de uma seção frente às solicitações de serviço;

Além disso, Os objetivos específicos são:

- Revisar a bibliografia, compilar os cálculos pertinentes, e fornecer um memorial detalhado do cálculo compilado;
- Elaborar o roteiro de cálculo e o algoritmo deste, verificando seções submetidas a Flexão Oblíqua Composta (FOC);
- Apresentar variáveis únicas (tipos) para as seções e suas armaduras, e trabalhar com procedimentos e rotinas que os utilizem;
- Aplicar os algoritmos em linguagem de programação orientada ao objeto;
- Elaborar exemplos de uso do programa, para demonstrar seu funcionamento;

## 1.3 METODOLOGIA DE PESQUISA

A pesquisa baseia-se no acervo de deduções e equações da bibliografia, principalmente, e com deduções e adaptações ao problema em questão, que é o de cálculo e verificação de uma seção sujeita a FCO.

Após a revisão bibliográfica inicial, os roteiros para cálculo, elaboração de algoritmo, aplicação em linguagem de programação e cálculo do exemplo para verificação deverão ocorrer simultaneamente. Certos detalhes e resoluções específicas podem ser necessários, incentivando partes da revisão bibliográfica a ocorrer durante qualquer etapa do trabalho, preenchendo cálculos imprevistos, devido à dificuldade do problema em questão.

A revisão bibliográfica é importante, pois possibilitará levar em conta o que já foi desenvolvido, reaproveitando o trabalho e continuando as pesquisas neste tema. Além disso, deverá apresentar ao leitor uma perspectiva de quais referências ele deve consultar em busca do tema desejado.

A elaboração dos algoritmos utiliza-se das bibliografias para escrever quais os passos para a resolução do problema, buscando uma maneira organizada e sucinta de apresentar o problema e sua possível solução. Os algoritmos definidos são úteis em todas as fases da elaboração do programa, inicialmente guiando no processo, norteando o programador até o

final da versão inicial. Mesmo depois do programa pronto, o algoritmo ainda é útil na releitura deste, pois permite que um usuário ainda não familiarizado com o programa possa reconhecer as etapas e entender o raciocínio do programador e o funcionamento do programa.

A escrita da linguagem de programação é a etapa que tem por foco transferir os conceitos apresentados em linhas do código de programação, que executa uma sequência de etapas com intuito de resolver o problema inicial. As linhas do programa não são apresentadas como integrantes do texto, pois o trabalho já apresenta os conceitos usados para obtê-las, portanto seria uma repetição do já exposto.

Assim, há um programa apresentado ao final do trabalho, que reúne os conceitos apresentados nesta dissertação. As linhas usadas para sua compilação também são disponibilizadas na íntegra. Ressalta-se que as principais rotinas do trabalho constam reunidas num arquivo importável, que tem por objetivo permitir fácil reuso e continuação do trabalho, com maiores explicações no capítulo 5.

Os métodos de cálculo nos estados-limites último e de serviço têm particularidades em comum, portanto espera-se que as rotinas sejam melhoradas continuamente, até o fim do trabalho e enquanto as simplificações e considerações usadas puderem ser mais sofisticadas e terem mais credibilidade. Os métodos usados serão explicitados e as rotinas, explicadas.

#### 1.4 ESTRUTURA DO TRABALHO

A divisão em capítulos segue o seguinte padrão:

No capítulo 2 são apresentadas as disposições e atribuições básicas, como as considerações recomendadas pela norma brasileira de estruturas de concreto ABNT NBR 6118:2014 e considerações de diversas bibliografias sobre o tema da F.O.C., explicando como considerar as solicitações na seção.

O capítulo 3 tem por objetivo explicar como considerar nos algoritmos as propostas e considerações do capítulo 2, apresentando de maneira mais didática do que simplesmente por linhas de programação – para que o leitor possa identificar o raciocínio geral para resolução de cada análise proposta (ELU, momento-curvatura ou ELS) e os passos a serem aplicados.

O capítulo 4 apresenta o resultado dos algoritmos apresentados, comparando-os com vários exemplos de distintas bibliografias, abordando várias possibilidades de uso para as rotinas apresentadas: exemplos em flexão reta, flexão oblíqua composta, relação momento-curvatura, verificações de estado-limite de serviço, uso de armadura ativa e concreto de resistência característica à compressão superior a 50 MPa e inferior a 90MPa.

O capítulo 5 tem foco na programação, apresentando as rotinas usadas nos cálculos no intuito de facilitar seu manuseio em futuros estudos. Apresenta-se a chamada de cada função e seus respectivos cálculos. Além disso, discorre-se sobre a exportação dos resultados do programa, com auxílio de um exemplo.

O capítulo 6 trata das considerações finais, apresentando os raciocínios de maneira resumida, além de discorrer sobre especificidades do cálculo em si: tanto sobre a interpretação normativa quanto sobre aspectos do cálculo numérico. Além disso, constam sugestões para próximos trabalhos.

## 2 REVISÃO BIBLIOGRAFICA E CONCEITOS FUNDAMENTAIS

Neste capítulo inicia-se uma breve revisão sobre trabalhos que abordam o tema da flexão oblíqua composta, destacando os aspectos mais relevantes e/ou que apresentam em comum com o presente trabalho. Devido à grande divergência existente quanto às abordagens usadas pelas bibliografias é difícil destacar as particularidades de cada obra em detalhe.

Além disso, deseja-se apresentar quais são as considerações admitidas para realização dos cálculos e adoções usadas no programa e nas rotinas elaboradas, para que seja mais fácil o entendimento do trabalho como um todo e principalmente das rotinas e do programa.

O entendimento das rotinas possibilita sua alteração e reuso, ou mesmo sua correção em caso de qualquer comportamento inesperado. E o entendimento do funcionamento do programa permite que o usuário avalie com mais segurança os resultados apresentados por ele, além de tornar os resultados mais compreensíveis já à primeira vista.

### 2.1 ALGUNS TRABALHOS

Neste trabalho deseja-se analisar seções que compõe elementos de barra. Tais elementos, segundo Mendes Neto (2010), são elementos tridimensionais com duas dimensões cujos comprimentos estão na mesma ordem de grandeza ( $b$  e  $h$  numa seção retangular, por definição), e a outra dimensão  $L$  de valor muito superior às duas outras. Assim, uma seção transversal corresponde à figura obtida após um corte perpendicular à maior dimensão da barra, isto é, seu eixo.

Ainda seguindo o raciocínio do mesmo autor, as peças são submetidas a uma variedade de carregamentos, que causam solicitações e apresentando deformações como resposta.

Nesse trabalho, deseja-se estudar as tensões denominadas normais, que de segundo a descrição de Fusco (1982), são as tensões que causam tensões perpendiculares ao plano das seções transversais dos elementos estruturais. O fato de tais tensões estarem em foco no trabalho indica que este não discorrerá sobre tensões tangenciais, como esforço de torção, por exemplo. Mais especificamente, deseja-se estudar o momento fletor aplicado na peça, que poderá ser aplicada em qualquer ângulo (momento usual e/ou momento transversal), numa seção também submetida à força axial.

E de acordo com a definição por Santos (1994), a flexão composta oblíqua (F.C.O.) ocorre quando o plano de ação do momento corta a seção segundo uma reta que não coincide com um eixo de simetria, ou quando o arranjo de barras não é simétrico em relação ao traço do plano do momento. Se as condições citadas forem respeitadas, trata-se de um problema de

flexão normal composta, que é um caso particular incluído na análise em flexão composta oblíqua.

Segundo Pallarés, Miguel e Fernández-Prada (2009) a F.C.O. ocorre na prática em diversas situações, por exemplo, pilares de canto da estrutura, ou pilares que sustentam lajes armadas em 2 direções, e seus estudos incluem diversas abordagens. O autor propõe um método numérico para resolução da flexão composta oblíqua.

Outro trabalho neste tema, ainda mais ousado, combina os esforços axiais, momentos, cisalhamento e torção para elementos de concreto armado e protendido, realizado por Gregori et al (2007). No trabalho, usa-se um modelo numérico em 3 dimensões usando método dos elementos finitos.

Uma forma de se resolver, usada no presente trabalho, é a usando os algoritmos com um conjunto de 3 equações de equilíbrio, fazendo uma varredura das tensões em diferentes possibilidades de carregamento da seção. Como descrito por Marino (1979), Fusco (1981) e Langendonck (1977), parte-se das condições de equilíbrio apresentadas pelas equações (2.1), (2.2) e (2.3).

$$N_d = \iint_{A_{cc}} \sigma_{cd} \cdot dX \cdot dY + \sum_{i=1}^n A_{si} \cdot \sigma_{sid} \quad (2.1)$$

$$M_{xd} = \iint_{A_{cc}} \sigma_{cd} \cdot X \cdot dX \cdot dY + \sum_{i=1}^n A_{si} \cdot \sigma_{sid} \cdot X_{si} \quad (2.2)$$

$$M_{yd} = \iint_{A_{cc}} \sigma_{cd} \cdot Y \cdot dX \cdot dY + \sum_{i=1}^n A_{si} \cdot \sigma_{sid} \cdot Y_{si} \quad (2.3)$$

Os autores descrevem que a posição desejada da linha neutra e o ângulo desta na seção são desconhecidos a princípio, e que o problema é resolvido percorrendo a seção, e aplicando as equações. A verificação e dimensionamento se tornam problemas com tantas variáveis, que usualmente, foram confeccionados vários ábacos para auxiliar os calculistas no dimensionamento de seções submetidas a tais carregamentos.

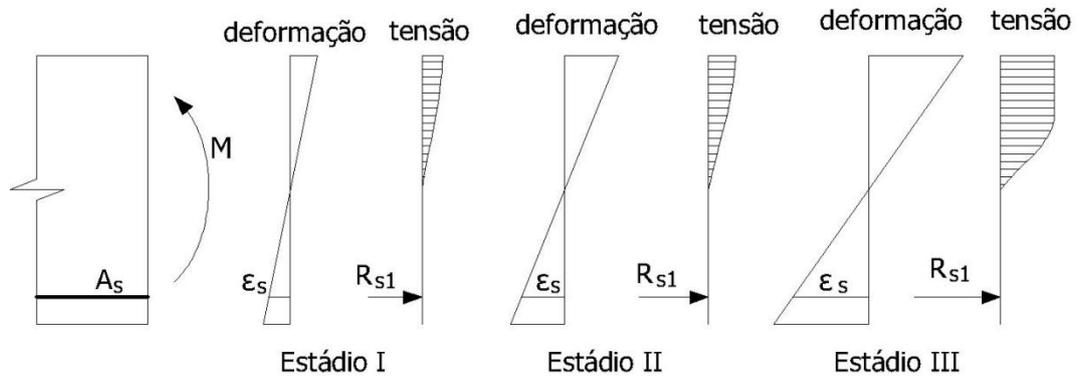
Finalmente, para que se analise as seções e suas respectivas solicitações normais obedecendo a norma brasileira vigente, a ABNT NBR 6118:2014, serão consideradas certas hipóteses básicas, detalhadas seguindo também a descrição de outras fontes, que trazem o mesmo raciocínio de maneira mais didática.

## 2.2 CONSIDERAÇÕES INICIAIS

Para que elas sejam consideradas, é desejável que se compreenda o comportamento de um elemento de barra submetido a tensões normais, neste caso momento fletor crescente. Diz-

se que, até seu colapso, a seção passará por estádios até que atinja sua ruptura, como descrito em Montoya, Meseguer e Cabré (1991) e Carvalho e Figueiredo Filho (2007) e apresentado simplificadaamente na Figura 1:

Figura 1 - Estádios de deformação



Adaptada de Montoya, Meseguer e Cabré (1991)

- Estádio I, estado elástico: o momento atuante é de pequena intensidade e não ultrapassa a resistência do concreto à tração, e considera-se:

- não há fissuras visíveis;
- a bibliografia considera nesses casos a tensão proporcional à deformação, pois nos trechos iniciais da curva de resistência do concreto o gráfico é aproximadamente linear: tal consideração não foi feita no presente trabalho;
- diagrama de tensão normal não é linear ao longo da seção, embora tal consideração seja razoável, tendo em vista o comportamento muito próximo ao linear;

- Estádio II puro, ou estado de fissuração, é o estágio em que o momento atinge valores tais que a as tensões de tração na borda inferior da seção passam a ultrapassar a resistência característica do concreto à tração, levando as considerações:

- apenas o aço passa a resistir às tensões de tração;
- as fissuras no concreto devido à flexão passam a ser visíveis;
- novamente, o diagrama tensão-deformação do concreto não foi considerado linear, apesar da variação extremamente suave, como mostrado pela bibliografia;

- Estádio III: trata-se do momento atuante na seção iminente à sua ruína. Tal condição permite que se considere:

- O concreto passa a plastificar a partir da deformação  $\varepsilon_{c2}$  (para  $f_{ck} \leq 50$  MPa, 2‰), alcançando a ruptura na deformação  $\varepsilon_{cu}$  (para  $f_{ck} \leq 50$  MPa, 3,5‰)
- as fibras de concreto que ultrapassam a deformação  $\varepsilon_{c2}$  passam a trabalhar em sua tensão máxima (tornando a parte superior do diagrama de tensões no concreto linear).
- a peça apresenta bastante fissuração, que devem se aproximar muito da linha neutra;
- supõe-se que a distribuição de tensões no concreto comprimido ocorra segundo a recomendação da ABNT NBR 6118:20014, isto é, seguindo a função neste trabalho citada, na seção 2.4.

Tendo em vista tal comportamento, apresentam-se as hipóteses de cálculo utilizadas, que tornam o cálculo utilizado neste trabalho possível, seguindo as recomendações da norma brasileira de estruturas de concreto, ABNT NBR 6118:2014:

- As seções transversais permanecem planas desde o início de seu carregamento até sua ruptura em estado-limite último, obedecendo a hipótese de Bernoulli: as deformações em cada ponto são proporcionais à sua distância até a linha neutra;

- Os materiais são idealmente solidários, de forma que as barras de armadura apresentem deformações iguais às do concreto adjacente;

- as tensões de tração no concreto são desprezadas, qualquer que seja o estágio de carregamento.

- O estado-limite último é caracterizado ao se alcançar os valores de deformação máximos: do concreto na borda mais comprimida e do aço na borda mais tracionada. Mais detalhes são apresentados na seção 2.3, em que são detalhados os domínios, isto é, estuda-se as possibilidades de conformação da seção quando em sua ruptura.

- As propriedades adotadas para os materiais são apresentadas em seções específicas, tais propriedades levam em conta as recomendações da norma brasileira de estruturas de concreto armado e os materiais disponíveis no mercado. Por definição, os valores de deformações limite padrões são:

- Para o concreto  $\varepsilon_{c,lim} = 3,5‰$  (compressão), apenas para  $f_{ck} \leq 50$  MPa;
- Para o aço, início do escoamento:  $\varepsilon_{yd} = 2,07‰$  (tração), e deformação na ruptura  $\varepsilon_{uk} = 10,0‰$  valor padrão para o aço CA-50.

Além disso, a linha-neutra da seção deve ser limitada como descreve a seção 14.6.4.3 da norma ABNT NBR 6118:2014. Sugere-se que o programa avise a altura da linha-neutra, ao se avaliar o ELU. A modificação propõe que o dimensionamento à flexão possua a linha-neutra mais limitada, buscando um comportamento mais dúctil da peça (portanto, menos

frágil). O programa não deve proibir o dimensionamento com  $x/d$  fora dos limites da norma, pois o usuário deve ter autonomia nas decisões, principalmente no caso de o usuário estar calculando uma peça sem focar o comportamento com foco na flexão, usando o programa apenas para não negligenciar o cálculo: por isso as ferramentas de cálculo devem apenas fornecer os dados e auxiliar o calculista, que sempre tem a palavra final à respeito das adoções admitidas.

Além disso, pode-se avaliar uma peça submetida predominantemente à compressão, porém com a presença de momentos fletores, com um comportamento típico de pilar, por exemplo: neste caso, o  $x/d$  provavelmente excederá os limites da norma para peças fletidas, e menos assim o programa não deve parar o cálculo, mas sempre avisar o calculista, alertando-o e tentando evitar imprevistos.

O citado novo limite para profundidade da linha neutra  $x/d$  para o dimensionamento de peças fletidas fazem com que a peça a ter um comportamento mais dúctil, maiores detalhes são explicados por Araújo (2009), que apresenta graficamente a capacidade plástica das peças dimensionadas sob diferentes profundidades de linha-neutra.

### 2.2.1 Convenção de sinais

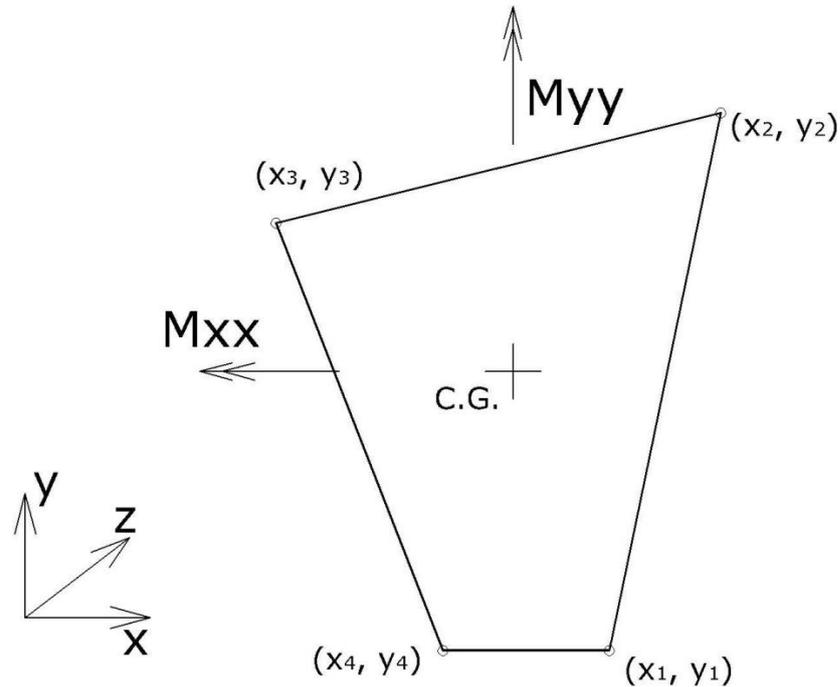
Seguindo as mesmas convenções usadas por Santos (1994) e Fusco (1982), serão usados os sinais positivo (+) para compressão e encurtamento; negativo (-) para tração e alongamento.

O momento aplicado em torno do eixo  $x$  é considerado positivo quando tracionar a parte inferior e comprimir a faixa superior, e o momento aplicado em torno do eixo  $y$  será positivo ao tracionar a parte esquerda e comprimir a parte direita – ver Figura 2.

A convenção de sinais escolhida para os momentos é apresentada na Figura 2 e a equação (2.4), mostrados a seguir.

$$\sigma(x, y) = \frac{N}{A} + \frac{M_{xx} \cdot y}{I_{xx}} + \frac{M_{yy} \cdot x}{I_{yy}} \quad (2.4)$$

Figura 2 – Convenção usada para os momentos na seção.



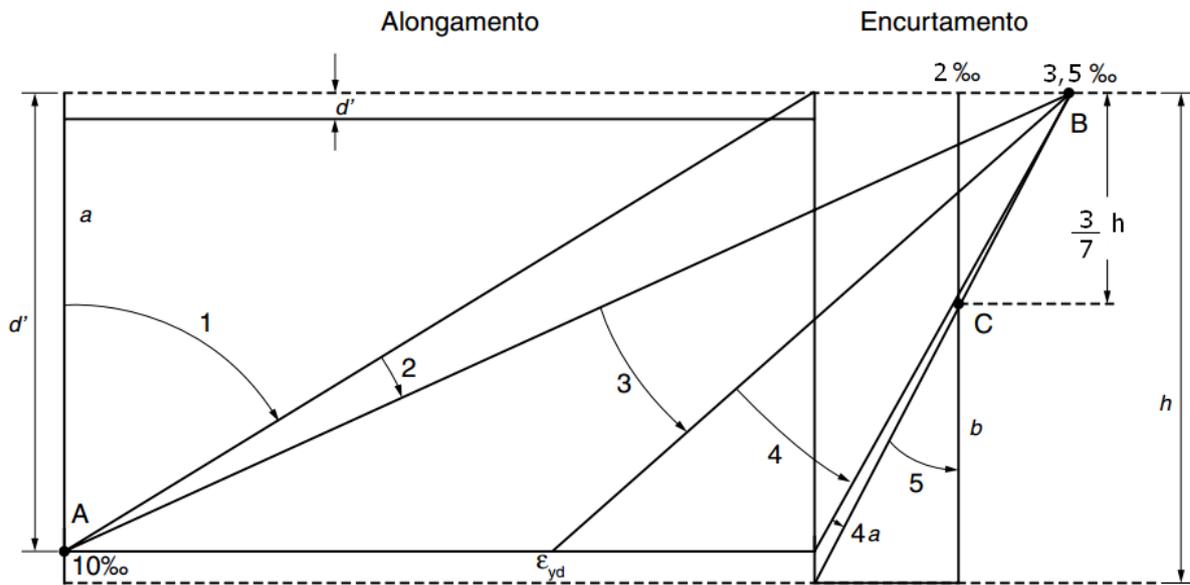
Além disso, cita-se a variável ângulo  $\alpha$ . Trata-se do giro da linha-neutra em sentido horário em relação à seção. Ou então, o giro da seção em torno de seu próprio centro na direção anti-horária, e mantendo a linha-neutra horizontal (paralela à  $x$ ). O ângulo da linha-neutra acompanha  $\alpha$ , o que é explicado mais detalhadamente no capítulo 2.5 e mostrado na Figura 13.

### 2.3 DOMÍNIOS DE DEFORMAÇÃO

Em seu estado-limite último, a seção é estudada em condições próximas à ruptura, em que a atuação das solicitações está na iminência de superar a capacidade resistente das peças, tais condições serão estudadas na presente seção. Salienta-se que esta abordagem e essas condições são diferentes daquelas abordadas para estado-limite de serviço: neste, as deformações podem estar em qualquer posição que não exceda aquelas citadas para estado-limite último, evidentemente.

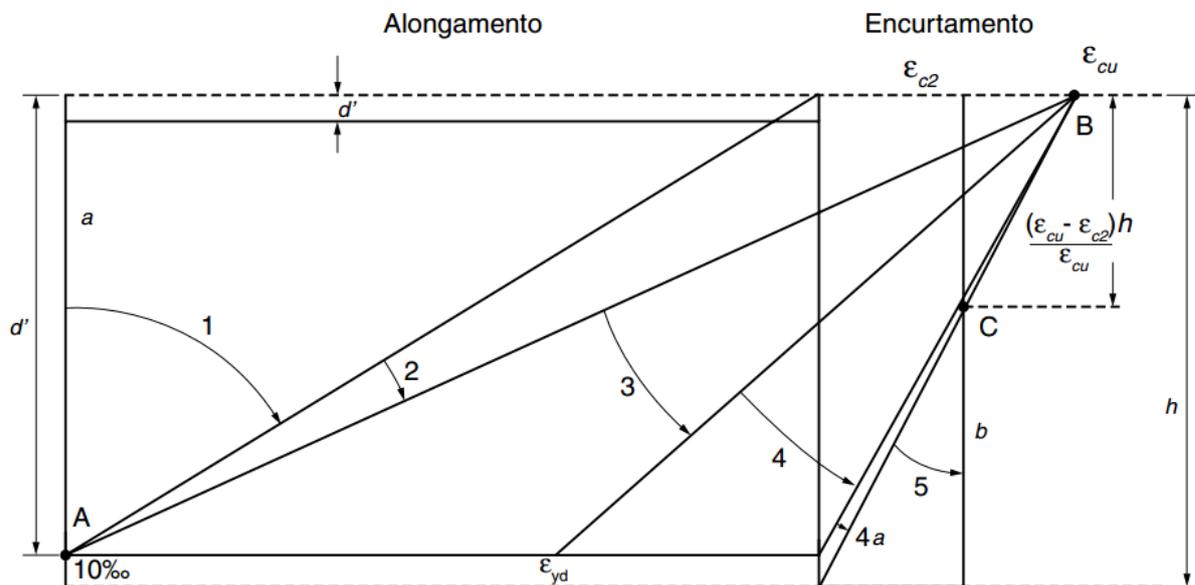
E como consta na NBR 6118:2014 e segundo Carvalho e Figueiredo Filho (2007), tal ruptura deverá ocorrer de forma que a seção apresente deformações correspondentes a um dos seguintes domínios, cujos valores são mostrados na Figura 3 e Figura 4. A diferença entre elas é dado pelo  $f_{ck}$  do concreto: se o  $f_{ck}$  usado for menor que 50 MPa, as considerações de materiais são as convencionais. Já se o  $f_{ck}$  for superior à 50 MPa, as deformações limites são diferentes, e até mesmo a função que descreve a tensão do concreto se altera, por isso é importante a diferenciação citada.

Figura 3 - Domínios de deformação em ELU para  $f_{ck} \leq 50$  MPa.



Fonte: Carvalho e Figueiredo Filho (2007).

Figura 4 - Domínios de deformação em ELU para  $f_{ck} > 50$  MPa.



Fonte: NBR 6118:2014.

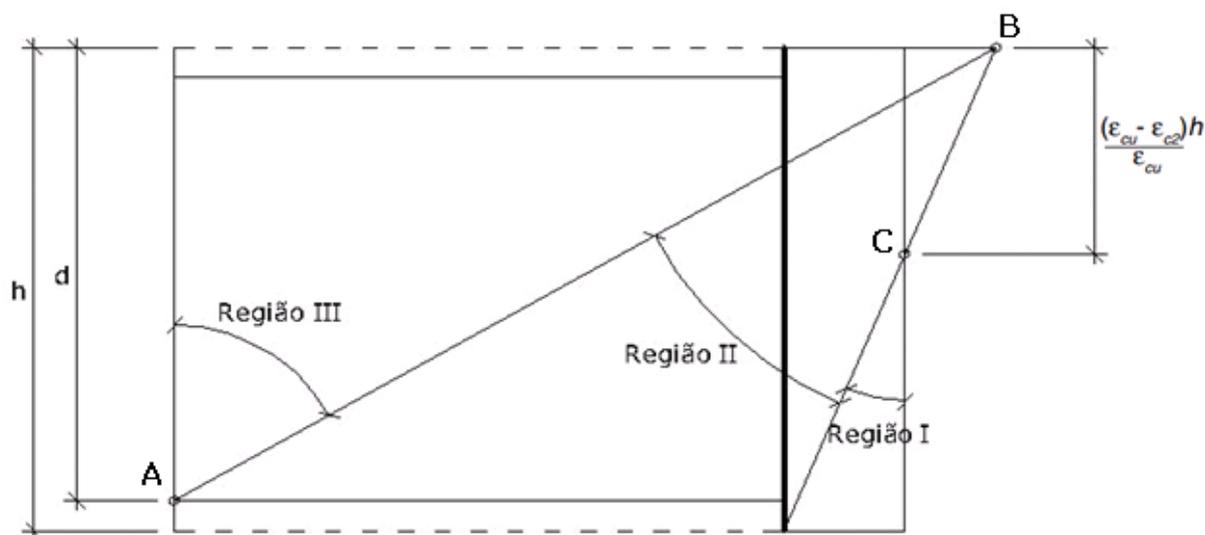
A partir da interpretação da Figura 3 e da Figura 4, pode-se definir como os limites de deformação no estado-limite último:

- Reta a: Tração uniforme
- Domínio 1: Tração não uniforme, sem compressão

- Domínio 2: Flexão simples ou composta sem ruptura à compressão do concreto com o máximo alongamento permitido na armadura;
  - Domínio 3: Flexão simples ou composta com simultaneidade de escoamento do aço tracionado e tensões de ruptura no concreto;
  - Domínio 4: Flexão simples ou composta tensões de ruptura no concreto, e antes da tensão de escoamento da armadura;
  - Domínio 4a: Flexão composta com armaduras comprimidas;
  - Domínio 5: Compressão não uniforme, sem tensões de tração;
- Reta b: Compressão uniforme.

Na programação, tais domínios serão considerados da mesma forma que propôs Santos (1994), apenas com nomenclatura diferente: verifica-se o ELU por 3 regiões, a região I, que corresponde ao ponto de ruína C; a região II, que corresponde ao ponto de ruína B; e a região III, que corresponde ao ponto de ruína A, como é mostrado na Figura 5. Porém, serão incluídos os critérios de deformações indicados pela ABNT NBR 6118:2014.

Figura 5 - As três regiões de ruptura.

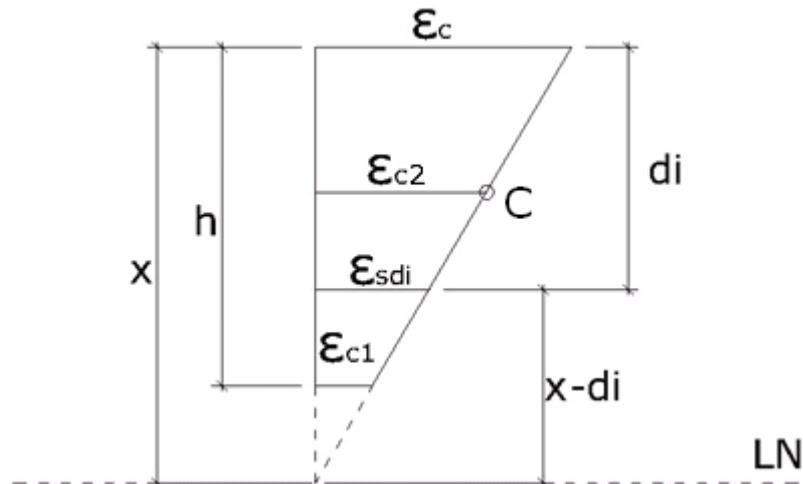


Fonte: Elaborado pelo autor.

A Região I caracteriza um ELU por ruptura de concreto, à compressão, e a deformação na seção segue como mostrado na Figura 6, a altura da linha neutra  $x$  é maior ou igual à altura da seção  $h$ , o encurtamento superior equivale a  $\epsilon_c$  e na borda inferior é definida por  $\epsilon_{c1}$ , e todas as retas que definem deformações giram em torno do ponto B, cuja deformação é 2‰, para  $f_{ck}$  menor ou igual a 50 MPa, e calculável pela expressão (2.5) caso

contrário, seguindo as recomendações da norma ABNT NBR 6118:2014. A Figura 6 também mostra a deformação  $\epsilon_{sdi}$ , que é a deformação do aço numa fibra qualquer a uma distancia  $d_i$  da fibra mais comprimida da seção.

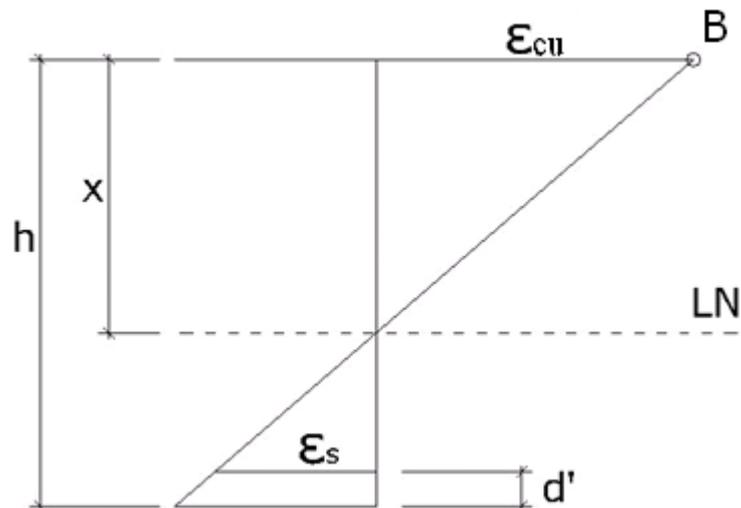
Figura 6 - Deformações na região I.



Fonte: Elaborado pelo autor

Já na região II, os diagramas de deformação em ruptura giram em torno do ponto B, cuja deformação é de  $\epsilon_{cu}$ , dada pela equação (2.6), e de encurtamento, e inclui os domínios 3, 4 e 4a. Nesta região a seção apresentará tração abaixo da linha neutra, e compressão acima. A Figura 7 mostra o diagrama de deformações, neste caso,  $x < h$ , como mostra a imagem.

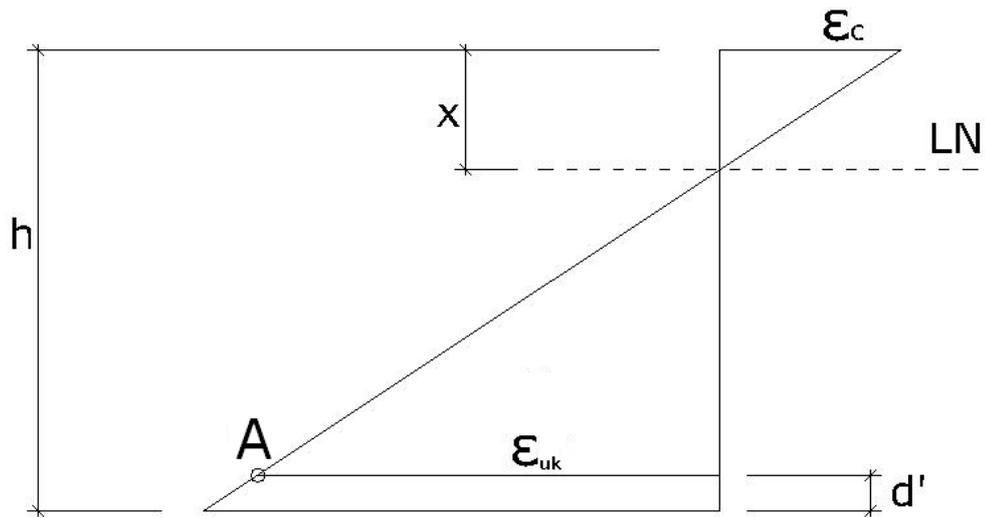
Figura 7 - Deformações na região II.



Fonte: Elaborado pelo autor

E na região III são incluídos os domínios 1 e 2. As deformações se dão em torno do ponto A, cuja deformação da armadura fixa-se em  $\epsilon_{yu}$  (com valor padrão 10‰), como mostrado na Figura 8, além disso, a linha neutra pode estar acima da seção, isto é, pode tratar-se de uma seção inteiramente tracionada.

Figura 8 - Deformações na região III.

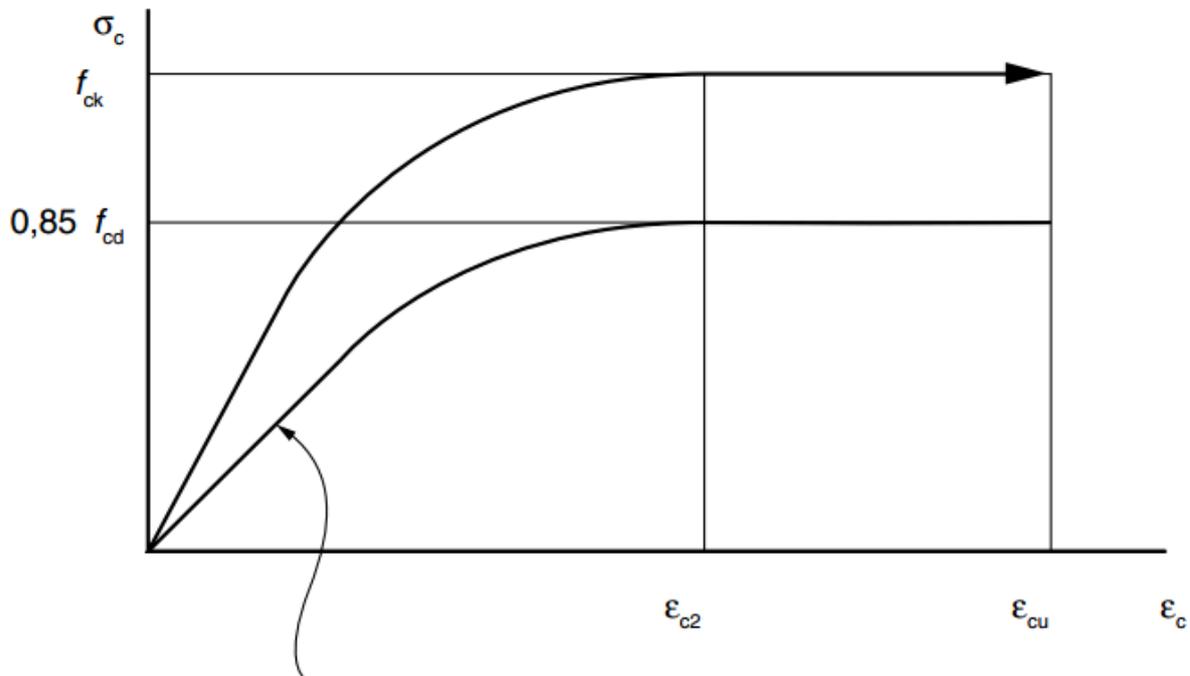


Fonte: Santos (1994)

#### 2.4 RELAÇÕES TENSÃO-DEFORMAÇÃO

Seguindo as recomendações da NBR 6118:2014, o comportamento do concreto em seu ELU será estimado pelo diagrama tensão-deformação mostrado na Figura 9:

Figura 9 - Diagrama tensão-deformação do concreto.



$$\sigma_c = 0,85 f_{cd} \left[ 1 - \left( 1 - \frac{\epsilon_c}{\epsilon_{c2}} \right)^n \right]$$

Para  $f_{ck} \leq 50$  MPa:  $n=2$   
 Para  $f_{ck} > 50$  MPa:  
 $n = 1,4 + 23,4 [(90 - f_{ck})/100]^4$

Fonte: NBR 6118:2014.

Tal diagrama será usado para determinação da tensão na seção de concreto em todos os domínios, exceto pelo domínio 1, que não sofre compressão.

Além disso, as expressões para deformação específica do encurtamento do concreto do início do patamar plástico  $\epsilon_{c2}$  e deformação específica de encurtamento do concreto na ruptura  $\epsilon_{cu}$  são descritas conforme a norma também, nas equações (2.5) e (2.6):

$$\epsilon_{c2} = \begin{cases} 2 \text{ ‰} & \text{se } f_{ck} \leq 50 \text{ MPa;} \\ 2 \text{ ‰} + 0,085 \text{ ‰} \cdot (f_{ck} - 50)^{0,53} & \text{se } 50 \text{ MPa} < f_{ck} \leq 90 \text{ MPa;} \end{cases} \quad (2.5)$$

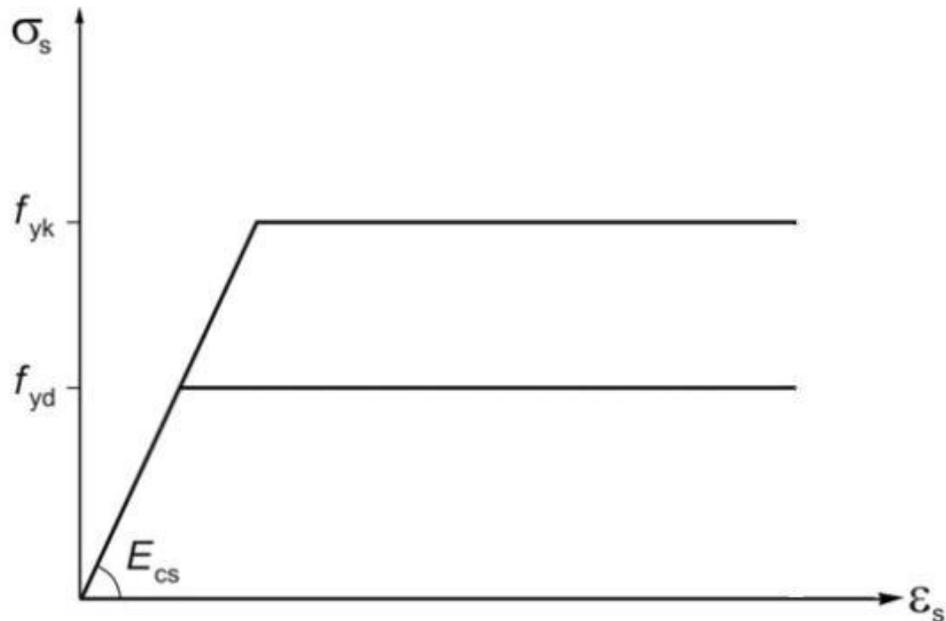
$$\epsilon_{cu} = \begin{cases} 3,5 \text{ ‰} & \text{se } f_{ck} \leq 50 \text{ MPa;} \\ 2,6 \text{ ‰} + 35 \text{ ‰} \cdot \left( \frac{90 - f_{ck}}{100} \right)^4 & \text{se } 50 \text{ MPa} < f_{ck} \leq 90 \text{ MPa;} \end{cases} \quad (2.6)$$

A resistência do concreto à tração não é considerada neste trabalho.

Já o aço tem diagrama tensão-deformação como mostrado pela Figura 10 e pela equação (2.7) e, que segue também as recomendações da NBR 6118:2014. Os sinais da deformação e da tensão permanecem iguais.

$$\sigma_s = \begin{cases} \varepsilon_s \cdot E_s & \text{Se } \varepsilon < \varepsilon_{yd}; \\ f_{yd} & \varepsilon_{yd} < \varepsilon; \end{cases} \quad (2.7)$$

Figura 10 - Diagrama tensão-deformação do aço.



Fonte: ABNT NBR 6118:2014.

A armadura resistirá a esforços de tração e compressão, que serão correlacionados usando se os sinais, positivo (+) e negativo (-), que serão levados em conta na aplicação das equações validas para cada região, e para cada diagrama e suas partes. O valor de  $\varepsilon_{yd}$  pode ser calculado pela equação (2.8).

$$\varepsilon_{yd} = \frac{f_{yd}}{E_{cs}} \quad (2.8)$$

Citando os valores padrão, referentes ao aço tipo CA-50, tem-se:

$$E_s = 210 \text{ GPa};$$

$$f_{yk} = 500 \text{ MPa};$$

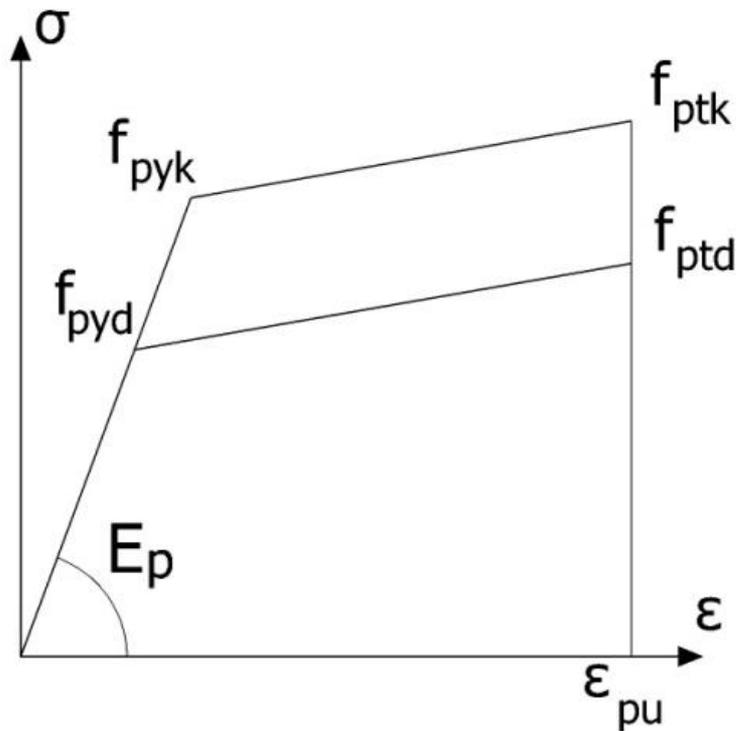
$$\gamma_s = 1,15;$$

$$\varepsilon_{yd} = 2,0704\%;$$

Além disso, deseja-se também abordar aplicação de armadura ativa, isto é, a protensão. O foco deste trabalho está nas peças sob pré-tração ou pós-tração aderente. No caso de uma análise de casos com pós tração não-aderente, recomenda-se considerá-la usando o carregamento uniformemente distribuído  $u$ , como descrito por Carvalho (2012).

Se tratando do método para se considerar armadura ativa neste trabalho, com o intuito de se obter as tensões da armadura ativa, se considera o pré-alogamento  $\varepsilon_p$  no cálculo das deformações da armadura. Além disso, as relações tensão-deformação seguem a conformação apresentada pela Figura 11 e representada matematicamente pelas equações (2.9) e (2.10), que serão utilizadas para obtenção das tensões da armadura, como descreve a norma de estruturas de concreto brasileira ABNT NBR 6118:2014.

Figura 11 – Relação tensão-deformação da armadura ativa



Fonte: ABNT NBR 6118:2014.

$$\sigma_p = E_p \cdot \varepsilon_t \quad \text{se } \varepsilon_t \leq \varepsilon_{pyd} \quad (2.9)$$

$$\sigma_p = f_{pyd} + \frac{(f_{ptd} - f_{pyd})}{(\varepsilon_{pu} - \varepsilon_{pyd})} \cdot (\varepsilon_t - \varepsilon_{pyd}) \quad \text{se } \varepsilon_{pyd} \leq \varepsilon_t \quad (2.10)$$

Ressalta-se que o gráfico (Figura 11) apresenta valores que não serão usados no cálculo realizado pelo presente trabalho, que são aqueles com sub-índice *k*. Os valores que interessam efetivamente para o cálculo apresentam o sub-índice *d*.

O cálculo da tensão será em função dos dados dos materiais, cujos valores padrão são aqueles apresentados pelo CP 190 RB, apresentados por Carvalho (2012) no capítulo 3, e revistos a seguir. Ressalta-se que os valores padrão possam ser alteráveis pelo usuário.

$$f_{pyd} = 1460 \text{ MPa};$$

$$f_{ptd} = 1626 \text{ MPa};$$

$$E_p = 200 \text{ GPa};$$

$$\varepsilon_{pu} = 35\text{‰};$$

Também inclui-se a variável  $\varepsilon_{pyd}$ , cujo valor será obtido pela divisão mostrada na equação (2.11). Na mesma equação, calcula-se o valor numérico de  $\varepsilon_{pyd}$  para o aço CP 190 RB.

$$\varepsilon_{pyd} = \frac{f_{pyd}}{E_p} \cong \frac{1460}{200000} = 7,3\text{‰} \quad (2.11)$$

Para calcular a deformação total da armadura, aplica-se a equação (2.12). O valor de  $\varepsilon_p$ , como descrito no início desta seção, é o valor do pré-alogamento. Tal valor deve ser fornecido pelo usuário, e como descreve Carvalho (2012) deverá incluir as perdas de protensão imediatas e com o passar do tempo. Já o valor de  $\varepsilon_s$  é o mesmo que seria utilizado para a armadura passiva se fosse o caso, isto é, é a deformação correspondente ao concreto adjacente.

$$\varepsilon_t = \varepsilon_s + \varepsilon_p \quad (2.12)$$

A partir do valor obtido para  $\varepsilon_t$ , pode-se calcular a tensão na armadura ativa, usando as equações (2.9) e (2.10).

## 2.5 VERIFICAÇÃO ESTADO-LIMITE ÚLTIMO

Conhecidas as características físicas e geométricas da seção, além da força normal atuante, pode-se calcular a envoltória dos momentos  $M_{xd}$  e  $M_{yd}$  que a seção resiste em cada direção. Assim são verificadas as seções sujeitas à F.C.O., gerando-se uma envoltória de possibilidades de esforços solicitantes, às quais a seção poderá resistir em estado-limite último.

As abordagens da bibliografia apresentam dois pensamentos diferentes referentes à resolução: o uso ou não de coeficientes adimensionais. Os coeficientes são  $\nu$ ,  $\mu_x$ ,  $\mu_y$  que relacionam os esforços e as dimensões da seção. Elas são apresentadas e discriminadas nas seções 4.2 e 4.3, ao serem apresentados os exemplos que utilizam esses coeficientes.

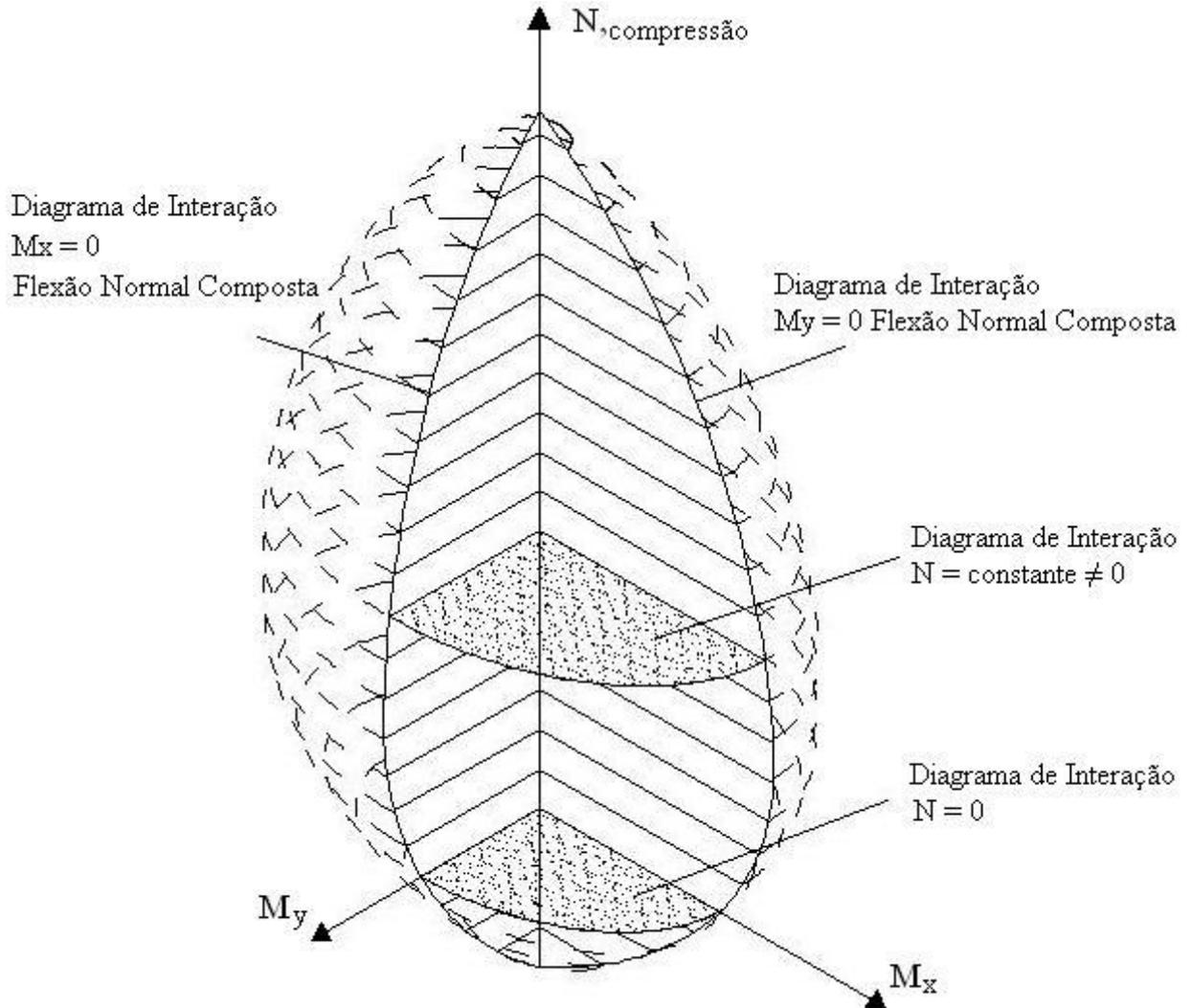
Marino (1979) tem foco nas variáveis adimensionais. A principal vantagem desta é que permite que o cálculo com velocidade razoável mesmo sem o uso de recursos computacionais. Porém, o cálculo requer consulta a ábacos e as seções devem seguir os padrões estabelecidos no respectivo ábaco, reduzindo a liberdade quanto à forma da seção e disposição da armadura. Fusco (1981) tem como foco uma abordagem sem o uso de tais variáveis adimensionais, e cita maneiras de se obter os diagramas de iteração com uso de uma sequência de operações (algoritmos).

Já Santos (1994) detalha várias formas de se analisar a questão considerando as duas abordagens diferentes, e seu uso computacional. No estudo de seções sob flexão oblíqua composta com uma abordagem computacional, recomenda-se fortemente consulta a esta bibliografia, que detalha vários métodos e rotinas extremamente úteis: tanto na abordagem usando variáveis adimensionais quanto na abordagem que não as utiliza.

O presente trabalho tem foco nos algoritmos computacionais. Logo, não se deseja estudar as variáveis adimensionais. Há bibliografias voltadas para essa abordagem, e grande parte delas trás ábacos para auxiliar os projetistas, como Montoya, Meseguer e Cabré (1991) e Venturini e Rodrigues (1987). Este último aborda os ábacos como tema principal.

Retornando à abordagem em foco, apresenta-se uma imagem de Fusco (1981) que facilita a visualização do problema e dos seus resultados, mostrado na Figura 12 a seguir, reforçando ser necessária a forma da seção transversal, a armadura desta e as características dos materiais usados.

Figura 12 - Superfícies de interação.

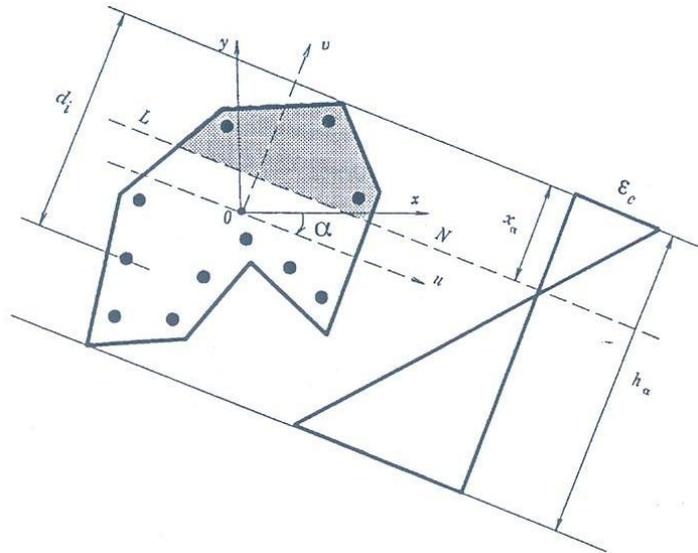


Adaptado de Fusco (1981).

De acordo com Santos (1994), ao se fixar um diagrama de deformações, conhece-se  $\epsilon$ , e pode-se calcular  $\sigma$ , constante em cada linha paralela à linha neutra (L.N.) da peça, que o autor trata como faixa.

Supondo uma seção poligonal qualquer, tendo-se o centro de ordenadas no centro geométrico desta, varia-se o ângulo  $\alpha$ , girando a seção em torno do centro geométrico desta, substituem-se as variáveis ortogonais  $x$  e  $y$  por outro eixo de coordenadas  $u$  e  $v$ , como mostrado na Figura 13.

Figura 13 - Seção genérica sob flexão composta oblíqua – coordenadas iniciais de referência  $xy$ .



Fonte: Santos (1994)

Com esta rotação  $\alpha$ , calcula-se as novas coordenadas para os vértices da seção e das barras. Adota-se uma posição da linha neutra  $x$ , e com esta, calcula-se as deformações e esforços em cada barra da seção, e calcula-se a parcela de força normal à qual o aço é responsável.

Então, com base no  $x$  adotado, delimita-se o polígono de compressão, e calcula-se tal esforço de compressão ao qual o concreto é responsável. Compara-se o esforço normal adotado com o esforço ao qual a seção de concreto e as barras de armadura passiva resistem. Conseguindo-se esforços normais resistentes e atuantes iguais (ou imensamente próximos), calculam-se os momentos resistentes em cada direção, para o concreto e aço, e obtém-se um par de momentos (segundo a convenção da Figura 13,  $M_x$  e  $M_y$ ) ao qual a peça resiste.

Caso os esforços normais não sejam compatíveis, altera-se a altura da linha neutra até que esta condição seja atendida. Uma vez atendida, varia-se o ângulo  $\alpha$ , e repete-se o processo até que os ângulos de  $0^\circ$  a  $360^\circ$  sejam verificados.

## 2.6 COMPORTAMENTO E VERIFICAÇÕES EM SERVIÇO

Além de se garantir a segurança da peça no estado-limite último, também deve-se verificar as condições da estrutura em serviço. Tais verificações dizem respeito à fissuração e deformação excessiva, diretamente ligadas à durabilidade e ao conforto em sua utilização, como descreve Carvalho (2012).

As estruturas de concreto protendido requerem verificações mais restritivas quanto à fissuração do que as estruturas de concreto armado segundo a norma de estruturas de concreto, tendo em vista a segurança contra a corrosão das armaduras ativas, pois estas apresentam tensões elevadas. Embora a fissuração não seja causa única, nem condição necessária, sua ocorrência acarreta num grande risco de degradação rápida do concreto superficial e da armadura, como descreve o mesmo autor.

As exigências da norma quanto à verificações de estado-limite de serviço com foco na fissuração seguem a indicação da Tabela 1 a seguir.

Tabela 1. Verificações de durabilidade relacionadas à fissuração.

Tipo Estrutural	Classe de Agressividade Ambiental (CAA)	Exigência
Concreto Armado (sem protensão)	CAA I	ELS-W ( $w_k \leq 0,4$ mm) – Combinação Frequente
Concreto Armado (sem protensão)	CAA II a CAA III	ELS-W ( $w_k \leq 0,3$ mm) – Combinação Frequente
Concreto Armado (sem protensão)	CAA IV	ELS-W ( $w_k \leq 0,2$ mm) – Combinação Frequente
Protensão Parcial	Pré-tração CAA I Pós-tração CAA I e II	ELS-W ( $w_k \leq 0,2$ mm) – Combinação Frequente
Protensão Limitada	Pré-tração CAA II Pós-tração CAA III e IV	(*)ELS-F. – Combinação Frequente (*)ELS-D – Combinação Quase-Permanente
Protensão Completa	Pré-tração CAA III e IV	(*)ELS-F – Combinação Rara (*)ELS-D – Combinação Frequente

Fonte: NBR 6118:2014.

Tendo em foco o uso da protensão e as verificações cabíveis, como apresenta a tabela anterior, os estados limites abordados neste trabalho são:

- Estado limite de abertura de fissuras (ELS-W): a dimensão da fissura deve respeitar certos limites;
  - Estado limite de formação de fissuras (ELS-F): a resistência à tração do concreto não deverá ser ultrapassada;
  - Estado limite de descompressão (ELS-D): a seção não apresentará tensões de tração;
- As definições na norma são apresentadas na seção 3.2.

Além destes 3 estados limites, deseja-se também apresentar o gráfico/relação momento-curvatura. Este é obtido ao comparar-se a curvatura da seção com o respectivo momento apresentado diante de tais deformações. Embora tenha foco no estado limite último, principalmente, a relação momento-curvatura é abordada nesta seção, pois as deformações na seção podem não estar em seu valor de ruptura.

Tal gráfico é importantíssimo na análise em concreto armado, podendo ser útil no estado-limite último e estado-limite de serviço. Por exemplo, na norma brasileira de estruturas de concreto, o método geral para o cálculo de instabilidade é o mais abrangente em termos de dimensionamento.

### 2.6.1 Gráfico Momento-Normal-Curvatura

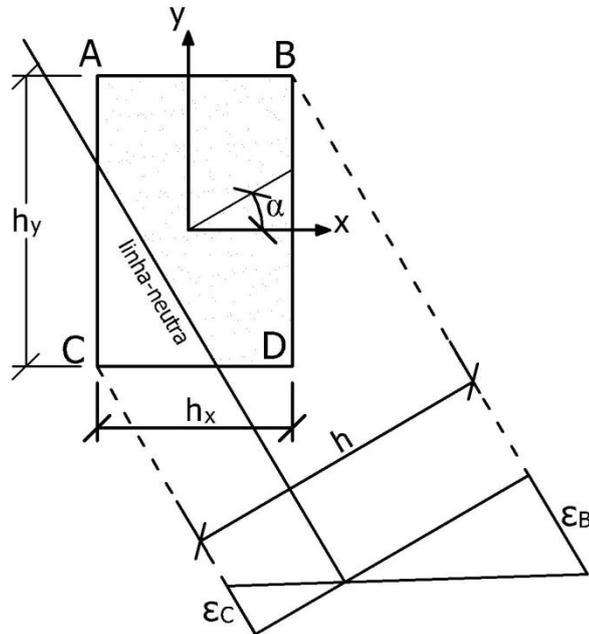
Na presente abordagem, um conceito importante é o da curvatura, que possibilitará cálculos mais complexos e importantes nas considerações das seções submetidas a solicitações normais. Reforça-se aqui a consideração de seção plana, essencial para esta seção em particular.

Inicialmente, segundo Leonhardt e Mönning (1979), que usa apresenta outra simbologia em seu livro, a curvatura  $1/r$  é expressa a partir dos valores absolutos das deformações nas fibras extremas  $\varepsilon_o$  e  $\varepsilon_u$  dividido pela distância  $d$  entre estas, como mostrado na equação (2.13).

$$\frac{1}{r} = \frac{\varepsilon_o - \varepsilon_u}{d} = \frac{M}{E \cdot I} \quad (2.13)$$

Já Fusco (1981) ao conceituar curvatura apresenta uma seção retangular com vértices A, B, C e D, do superior ao inferior, da esquerda para a direita (ver Figura 14). Com tais valores, o mesmo autor demonstra as equações (2.14) e (2.15), apenas usando denominações diferentes para os pontos, mas com o mesmo significado final.

Figura 14 – Exemplo de conformação de curvatura na seção e convenção indesejada para  $\alpha$



$$\frac{1}{r_x} = \frac{\varepsilon_A - \varepsilon_B}{h_x} = \frac{\varepsilon_C - \varepsilon_D}{h_x} \quad (2.14)$$

$$\frac{1}{r_y} = \frac{\varepsilon_A - \varepsilon_C}{h_y} = \frac{\varepsilon_B - \varepsilon_D}{h_y} \quad (2.15)$$

E ainda apresenta as relações (2.16), (2.17) e (2.18), muito úteis entre as variáveis, sendo  $\alpha$  um ângulo qualquer de giro na seção, como mostrado na Figura 14.

$$h_x \cdot \frac{1}{r_x} + h_y \cdot \frac{1}{r_y} = (h_x \cdot \cos \alpha + h_y \cdot \sin \alpha) \cdot \frac{1}{r_\alpha} \quad (2.16)$$

$$\frac{1}{r_x} = \frac{1}{r_\alpha} \cdot \cos \alpha \quad (2.17)$$

$$\frac{1}{r_y} = \frac{1}{r_\alpha} \cdot \sin \alpha \quad (2.18)$$

Porém, destaca-se que essas equações serão reescritas ao serem aplicadas neste trabalho, pois o ângulo  $\alpha$  apresentado na Figura 14 diverge da abordagem desejada ao  $\alpha$  no presente trabalho (ver Figura 13). Pode-se converter as equações usando um ângulo intermediário  $\beta$  ou ainda alterar a referência de  $\alpha$ . No presente trabalho adota-se alterar a referência de  $\alpha$ : o proposto por Fusco (1981) é que ele inicie paralelo ao eixo  $x$  e seja crescente no sentido anti-horário. No presente trabalho,  $\alpha$  inicia paralelo ao eixo  $y$  e cresce no sentido horário, assim como as rotinas do capítulo 0, mais especificamente na seção 3.1.3, seguindo o padrão adotado por Santos (1994).

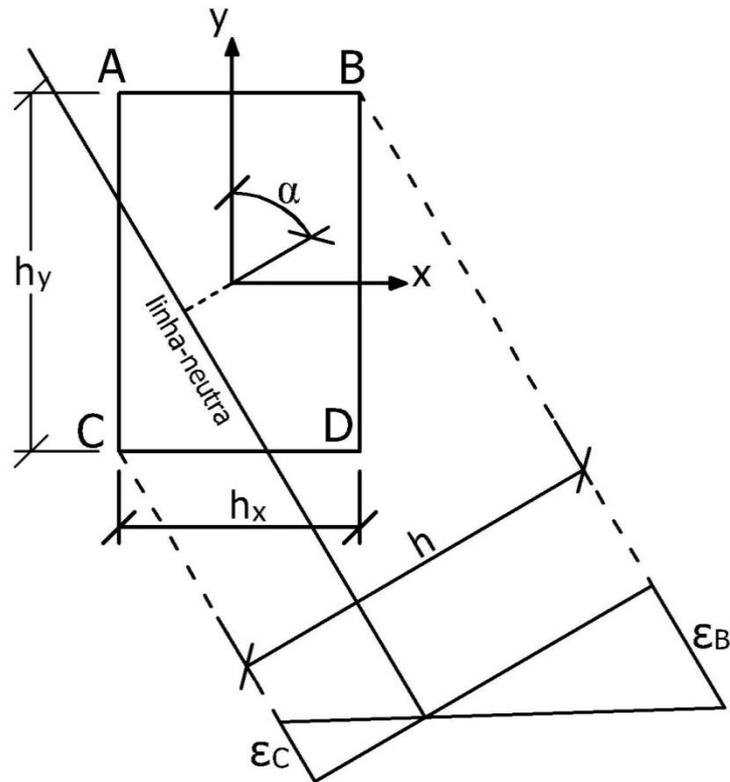
Logo, seguindo a abordagem proposta, substitui-se (2.16), (2.17) e (2.18) por (2.19), (2.20) e (2.21), e a Figura 14 pela Figura 15.

$$h_x \cdot \frac{1}{r_x} + h_y \cdot \frac{1}{r_y} = (h_x \cdot \sin \alpha + h_y \cdot \cos \alpha) \cdot \frac{1}{r_\alpha} \quad (2.19)$$

$$\frac{1}{r_x} = \frac{1}{r_\alpha} \cdot \sin \alpha \quad (2.20)$$

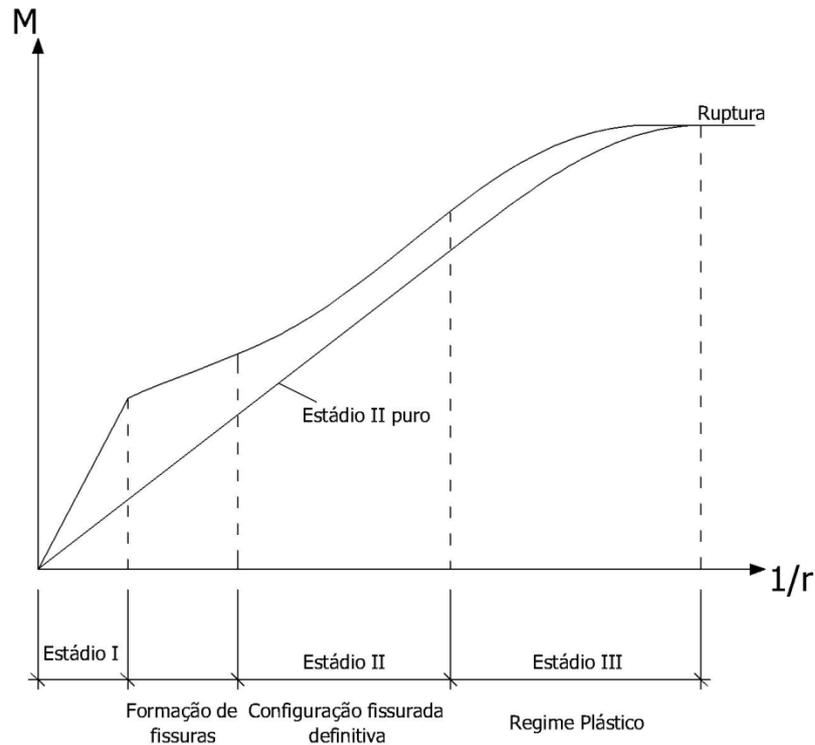
$$\frac{1}{r_y} = \frac{1}{r_\alpha} \cdot \cos \alpha \quad (2.21)$$

Figura 15 – Exemplo de conformação de curvatura na seção e convenção usada para  $\alpha$



Um gráfico que apresenta a relação momento-curvatura e aproxima o seu comportamento considerando a seção dentro do estágio I, II e III é apresentada por Leonhardt e Mönnig (1979), mostrado na Figura 16, mostrando a conformação do momento-curvatura da peça em 2 situações: em sua conformação real, em que o concreto resiste às solicitações de tração até certa magnitude (isto é, dentro do estágio I); e desprezando a resistência a tração do concreto, (estádio II puro)

Figura 16 – Relações momento-curvatura real e supondo estágio II puro.

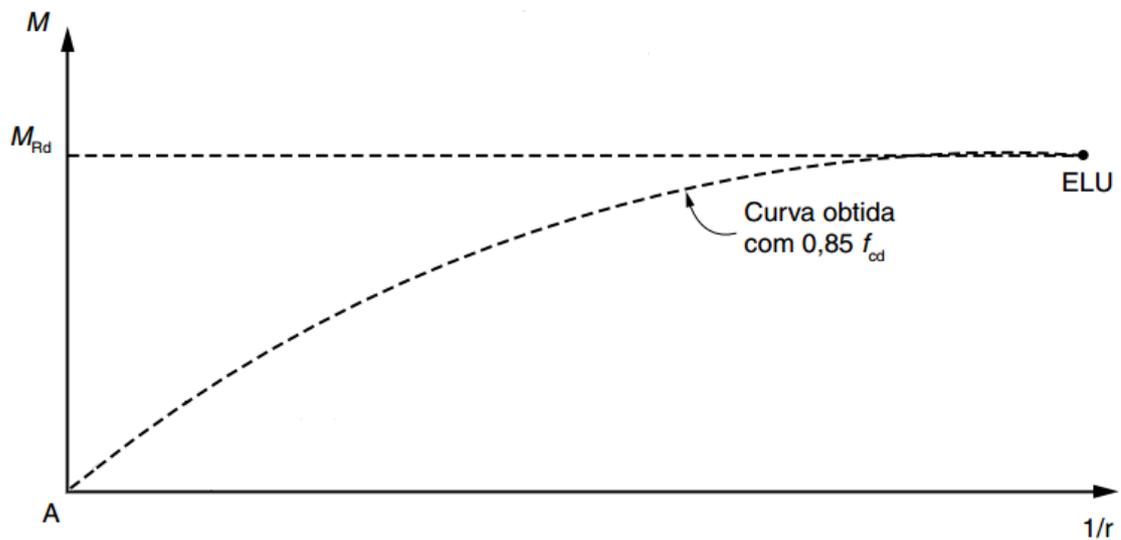


Adaptado de Leonhardt e Mönnig (1979).

Salienta-se que no presente trabalho, o gráfico apresenta a conformação da curva inferior, cuja bibliografia se refere como “Estádio II puro”, assim como destaca a seção 2.4.

Ainda tratando sobre as relações momento-curvatura segundo a bibliografia, segue na Figura 17 o indicado segundo a norma ABNT NBR 6118:2014 como o aspecto da relação momento-curvatura.

Figura 17 – Relação momento-curvatura: aspecto segundo a norma



### 2.6.2 Verificações de estado limite de abertura de fissuras

Seguindo as recomendações da seção 17.3.3.2 da norma, o critério para se calcular a abertura de fissuras numa estrutura segue as equações (2.22) e (2.23).

$$w_k = \text{Menor} \left\{ \begin{array}{l} \frac{\phi_i}{12,5\eta_1} \frac{\sigma_{si}}{E_{si}} \frac{3\sigma_{si}}{f_{ctm}} \\ \frac{\phi_i}{12,5\eta_1} \frac{\sigma_{si}}{E_{si}} \left( \frac{4}{\rho_{ri}} + 45 \right) \end{array} \right. \quad \begin{array}{l} (2.22) \\ (2.23) \end{array}$$

Em que:

$w_k$  = abertura de fissuras;

$E_{si}$  : Módulo de elasticidade do aço da barra considerada, de diâmetro  $\phi_i$ ;

$\phi_i$  : Diâmetro da barra que protege a região de envolvimento considerada;

$\eta_1$  : Coeficiente de conformação superficial da barra;

$\rho_{ri}$  : taxa de armadura ativa aderente ou passiva em relação a área de envolvimento ( $A_{cri}$ );

$\sigma_{si}$  : tensão de tração no centro de gravidade da armadura considerada, calculada no estado II;

O coeficiente de conformidade superficial  $\eta_1$  da armadura passiva e ativa, segundo a NBR 6118:2014, é de 1,0 para barras lisas, 1,4 para barras entalhadas e 2,25 para barras nervuradas (tabela 8.3 do texto da norma). Além disso, na seção 9.3.2.2 constam para armaduras ativas: 1,0 para fios lisos, 1,2 para cordoalhas de 3 e 7 fios, e 1,4 para fios dentados.

No presente trabalho, calcula-se apenas o primeiro termo da equação de abertura de fissuras, isto é, apenas a equação (2.22), devido às variáveis usadas por cada uma das equações: os dados da primeira podem ser calculados usando os dados das rotinas elaboradas, enquanto a equação (2.23) requer o cálculo das áreas de abrangência, cuja alternativa mais viável e segura aparenta ser um sistema que análise a seção graficamente.

Mesmo assim, o programa e as rotinas fornecem a tensão nas barras, facilitando o cálculo da segunda equação, basta que se calcule  $A_{cri}$  e  $\rho_{ri}$ , pois os outros fatores são constantes ou valores bastante simples, com exceção da tensão, que como dito, é fornecida pelas rotinas do programa.

### 2.6.3 Verificações de estado limite de formação de fissuras

Outra verificação de estado limite de serviço é a de formação de fissuras.

Segundo a norma ABNT NBR 6118:2014, para atender a este estado limite de formação de fissuras, um procedimento para realizar a verificação consiste em comparar o momento atuante na seção com o momento de fissuração, isto é, um momento a partir do qual a seção fissa.

Para calcular o momento de fissuração, a norma apresenta a equação (2.24)

$$M_r = \frac{\alpha \cdot f_{ct,inf} \cdot I_c}{y_t} \quad (2.24)$$

Em que:

$f_{ct,inf}$  : Minoração de  $f_{ct,m}$ , expresso em Megapascal (usado no estado-limite de formação de fissuras, para analisar o estado-limite de deformação excessiva é usado outro valor);

$f_{ck}$  : Resistência característica à compressão do concreto, em expresso em Megapascal;

$M_r$  : Momento de fissuração da seção;

$\alpha$  : Fator que correlaciona aproximadamente a resistência à tração na flexão com a resistência à tração direta e equivale a 1,2 para seção T ou duplo T, 1,3 para seções I ou T invertido e 1,5 para seções retangulares.

$I_c$  : Momento de inércia da seção de concreto;

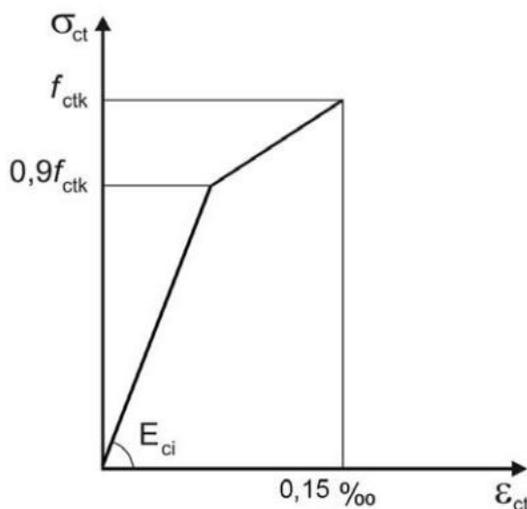
$y_c$  : Distância do centro de gravidade até a borda inferior da seção;

Porém tal consideração neste trabalho não será realizada, devido à suas limitações: o coeficiente  $\alpha$  supõe formas fixas para a seção transversal, limitação esta indesejada neste trabalho.

Além disso, a forma se assemelha à formula da flexão (embora a norma não detalha sobre sua origem) e a consideração de um comportamento linear dos materiais não é feita neste trabalho e tal consideração é pré-requisito para aplicação da formula da flexão.

Assim sendo, deseja-se avaliar a ocorrência ou não da fissuração por uma perspectiva de tensões e deformações. Para apresentar o critério usado neste trabalho, apresenta-se o gráfico tensão-deformação para concreto tracionado segundo a norma ABNT NBR 6118:2014, na Figura 18.

Figura 18 – Diagrama tensão-deformação do concreto à tração.



Mesmo assim, não é considerada no trabalho a resistência do concreto à tração. Apresenta-se o gráfico com intuito de justificar a deformação limite adotada. Com base na Figura 18, a deformação limite para o estado-limite de formação de fissuras é de 0,15‰. Sugere-se que o valor possa ser alterado, para que este possa se adequar a possíveis atualizações neste estado-limite, ou ainda para que o usuário use um valor que julgue mais condizente com esta verificação.

Um outro valor que pode ser razoável nesta análise é o  $f_{ct,inf} / E_{ci}$  ou ainda levar em conta apenas a primeira linha na Figura 18:  $0,9 f_{ct,inf} / E_{ci}$ . Tais valores podem representar a deformação do concreto na ruptura, dependendo das adoções consideradas pelo calculista.

#### 2.6.4 Verificações de estado limite de descompressão

Este estado limite impõe uma seção totalmente sem tração, em que os pontos mais tracionados apresentem tração nula, sendo uma verificação usual para concreto protendido.

Na abordagem deste trabalho, essa verificação pode ser realizada avaliando-se as deformações nos extremos dos pontos da seção, avaliação esta especialmente fácil pois a linha neutra sempre é paralela ao eixo das abscissas  $x$  (após a rotação da seção), e a deformação no topo da seção sempre é mais comprimida que a deformação na borda inferior. Logo, avalia-se se deformação na borda inferior assume valor negativo (deformação de tração) ou positivo (deformação de compressão).

Caso tal deformação seja de compressão ou nula ( $\epsilon \geq 0$ ), tal verificação é atendida.

### 3 ROTINAS DE CÁLCULO

Apresentam-se as rotinas em diversas seções, cada uma explicando uma etapa no cálculo, e discriminada de acordo com sua posição num programa final. Porém, de acordo com a análise do programador e a forma de interpretação do problema, a ordem de aplicação destas pode alterar.

Pode-se destacar um conjunto de operações, que se repete em várias partes de todas as análises, e agrupa várias rotinas, que são as operações que calculam as solicitações resistentes em função das deformações atribuídas à seção. Essas operações são repetidas em todas as análises (ELU, Relação Momento-Curvatura e ELS). Por isso, o tratamento matemático usado considera as solicitações como função das deformações, e aplica-se os métodos numéricos para se obter raízes de funções não lineares.

As últimas rotinas são as aplicações dos algoritmos finais: isto é, executam as operações mostradas no decorrer desta seção para calcular um dado final. Por exemplo, a seção 3.6, estado-limite último, apresenta o algoritmo que apresenta os momentos resistentes na seção para determinada força normal, com auxílio das rotinas mostradas.

#### 3.1 PREPARAÇÃO DOS DADOS

Com as coordenadas do polígono de concreto, coordenadas das armaduras e da resistência dos materiais, fornecidas pelo usuário, pode-se calcular os diagramas de resistências da seção em estado-limite último e seu comportamento (momento-curvatura) sob certa força axial. Para isso os dados fornecidos pelo usuário serão ajustados de forma a facilitar seu manuseio, visto que os cálculos, métodos e rotinas utilizados são relativamente complexos e/ou extensos. Grande parte das rotinas aqui mostradas foi idealizada por Santos (1994), ou seguindo as orientações de Fusco (1981) e Marino (1979), ou ainda elaborada pelo próprio autor com base na bibliografia e nas rotinas.

Primeiramente, calcula-se o centro geométrico do polígono que define a seção de concreto. Os cálculos de momento são realizados considerando o braço de alavanca em relação a este ponto, isto é, as excentricidades tem como origem o centro geométrico do polígono de concreto.

Para que o cálculo do centro geométrico calcula-se primeiramente a os momentos estáticos da seção de concreto, que dividido pela área resulta na coordenada de centro, como apresenta a seção 3.1.1.

Dando continuidade, ao se obter as coordenadas do centro geométrico da seção, faz-se a translação da origem do eixo de coordenadas, como consta mais detalhadamente na seção 3.1.2.

Além disso, para calcular as forças e momentos em diferentes ângulos, é necessário que a linha neutra gire em relação à seção do elemento estrutural, processo este mais detalhado na rotina descrito na seção 3.1.3.

Com os pontos situados na posição desejada, calculam-se as forças normais máxima e mínima da seção, considerando apenas carregamento axial, para saber quais diagramas de força são possíveis de serem calculados.

Primeiramente, gira-se a seção para o ângulo desejado, e com ela fixada neste ângulo, calcula-se as características da seção nesse ângulo, como os pontos máximos e mínimos da seção e armadura e a altura da seção. Tais cálculos devem ser feitos cada vez que se gira a seção novamente.

Para cada ângulo, fixa-se um diagrama de deformações inicial, e são calculadas as forças resultantes na seção para este diagrama de deformações. Varia-se o diagrama de deformação, e calcula-se a força resultante novamente, sempre comparando com a força normal procurada. A cada diagrama de deformações, calcula-se o valor de  $x$  (profundidade da linha neutra, em relação ao ponto mais alto da seção), o valor de  $x_{cc2}$ , que é o tamanho da faixa de concreto cuja tensão é uma função, apresentado com mais detalhes na seção 4.1.3.

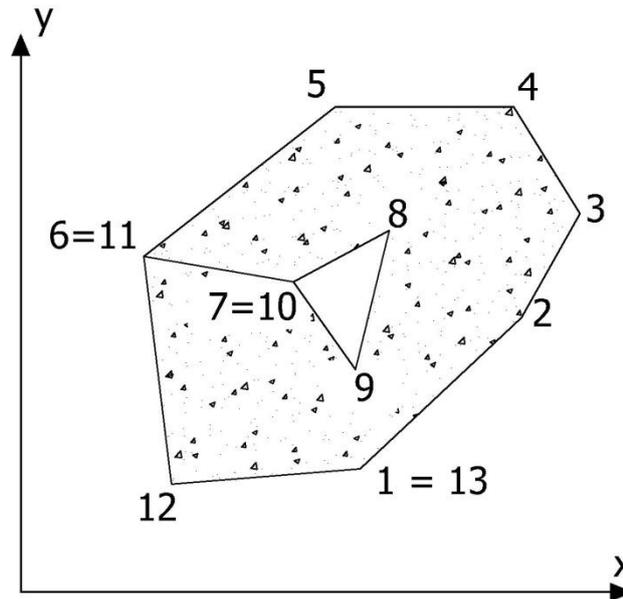
### **3.1.1 Características geométricas**

Nesta etapa inicial do algoritmo, calculam-se certas características geométricas da seção. No algoritmo do programa, a característica calculada é o centro geométrico da seção de concreto: logo, não se consideram nesta etapa a armadura passiva ou ativa.

As formulas usadas seguem a indicação de Santos (1994), que em seu livro apresenta uma rotina que as executa.

Apresenta-se o cálculo requerido para obter-se as coordenadas do centroide da seção. Para isso, considera-se uma seção transversal plana e poligonal, tendo como exemplo o apresentado pela Figura 19.

Figura 19 – Exemplo de figura plana definida por uma poligonal.



A poligonal deve ser indicada pelo usuário pela inserção de seus respectivos pontos, de modo que os pontos externos deverão ser indicados no sentido anti-horário, que simbolizam área positiva; já eventuais vazios deverão ser indicados por pontos no sentido horário, assim como apresenta a Figura 19.

Com os pontos definidos, calcula-se  $aux_i$  (3.1), que compõe os cálculos da área, em (3.2), e dos momentos estáticos em torno de x e y, usando (3.3) e (3.4). Finalmente, as coordenadas  $x_{CG}$  e  $y_{CG}$  do centroide são calculadas pelas equações (3.5) e (3.6).

$$aux_i = x_i \cdot y_{i+1} - x_{i+1} \cdot y_i \quad (3.1)$$

$$A_c = \frac{1}{2} \cdot \sum_{i=1}^{nv} aux_i \quad (3.2)$$

$$S_{xx} = \frac{1}{6} \cdot \sum_{i=1}^{nv} (y_i + y_{i+1}) \cdot aux_i \quad (3.3)$$

$$S_{yy} = \frac{1}{6} \cdot \sum_{i=1}^{nv} (x_i + x_{i+1}) \cdot aux_i \quad (3.4)$$

$$x_{CG} = \frac{S_{yc}}{A_c} \quad (3.5)$$

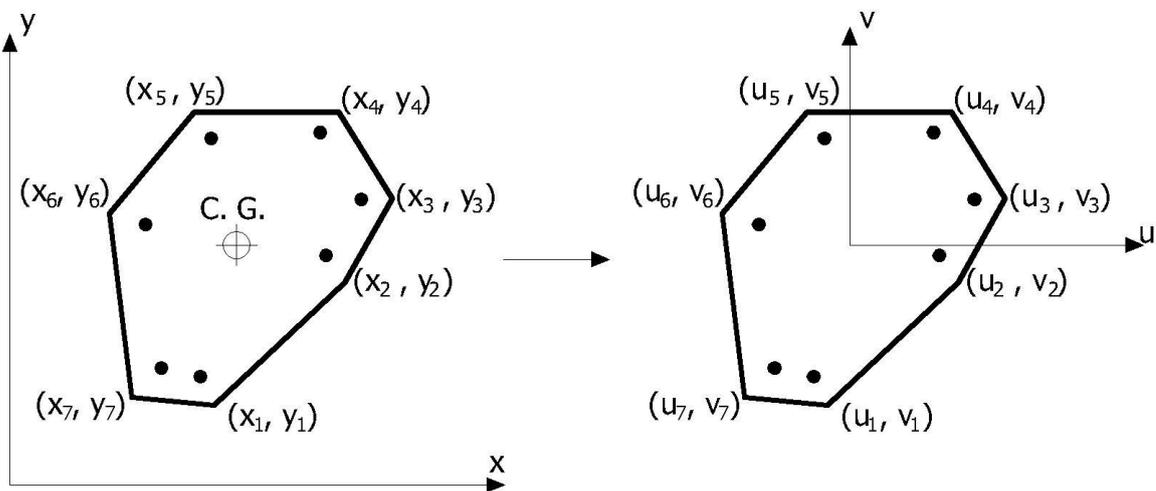
$$y_{CG} = \frac{S_{xc}}{A_c} \quad (3.6)$$

Salienta-se que o polígono não necessariamente deverá ser fechado (ter o último e primeiro ponto iguais). Na Figura 19 apresenta-se tais pontos coincidindo, porém pode-se considerar que o último ponto é o 12: ao calcular-se o  $aux_{12}$ , se faz necessário um  $y_{13}$ , que deverá ser igual a  $y_1$ , como descrito por Santos (1994).

### 3.1.2 Translação da origem do eixo de coordenadas

A translação do eixo de coordenadas não é um pré-requisito no cálculo, mas é feito para se facilitá-lo, principalmente do momento: é desejável mover o eixo de referência, de modo ao centro geométrico da seção coincidir com o centro do plano cartesiano  $u v$ , exemplificado na Figura 20, de eixos  $x y$  para  $u v$ .

Figura 20 – Translação de coordenadas em seção genérica.



A expressão usada no cálculo é extremamente intuitiva, ainda assim é apresentada segundo as expressões (3.7) e (3.8).

$$u_i = x_i - x_{CG} \quad (3.7)$$

$$v_i = y_i - y_{CG} \quad (3.8)$$

### 3.1.3 Rotação da origem do eixo de coordenadas

Nessa etapa de cálculo a linha neutra deve ser inclinada por um ângulo qualquer: para isso, será necessário girar os pontos em relação ao plano cartesiano também: o centro do plano cartesiano está posicionado no centro geométrico da seção e mesmo depois de girados ainda deverão estar no centro geométrico, assim como mostra a Figura 13.

Uma recomendação é que haja dois vetores que guardem as coordenadas da poligonal, tanto para o concreto quanto para a armadura. Um vetor que contém os pontos da poligonal original, que sempre fique na posição inicial, e outro vetor, de uso temporário e cujos dados

sempre sejam gerados a partir do vetor inicial, para evitar acumulação e propagação de erros numéricos. Além disso, quando se trata de vetor, não se trata de uma variável vetorial, mas de um conjunto de dados, um vetor na linguagem computacional.

Considerando que a rotação ocorrerá em torno de um ponto coincidente com a origem do eixo de coordenadas cartesianas, as expressões são apresentadas em (3.9) e (3.10), de acordo com Santos (1994).

$$u_i = x_i \cdot \cos \alpha - y_i \cdot \sin \alpha \quad (3.9)$$

$$v_i = x_i \cdot \sin \alpha + y_i \cdot \cos \alpha \quad (3.10)$$

### 3.2 CÁLCULO DA FORÇA E MOMENTOS DO CONCRETO

Na entrada de dados do programa, é necessária a entrada de um polígono que mostre a forma da seção de concreto. Esse recurso busca tornar as rotinas abrangentes em sua análise, isto é, não apenas analisar seções retangulares, seções T, e etc. Embora tais seções sejam as mais utilizadas pelo mercado e de execução mais conhecida, procura-se oferecer a possibilidade de seções quaisquer, desde que sejam representáveis por um polígono.

Assim, o usuário também pode aproximar seções de formas não poligonais: aproximando curvas em vários segmentos de retas secantes ao trecho, por exemplo. Mesmo um número elevado de pontos não deverá tornar a análise dispendiosa, tendo-se em vista a imensa quantidade de dados que os computadores processam nos dias de hoje.

No raciocínio adotado, a linha neutra separa a parte tracionada da parte comprimida da seção. A parte superior é a comprimida e a inferior é a tracionada, e apenas tensões de compressão são admitidas no concreto na presente análise.

Segundo as recomendações da norma ABNT NBR 6118:2014, como já apresentado, pode-se usar o seguinte raciocínio para equacionar o problema, o que se faz necessário, para que seja possível implementá-lo computacionalmente, mostrado pela função (3.11):

$$\text{Se } \varepsilon \leq 0 \rightarrow \sigma_c = 0$$

$$\text{Se } 0 < \varepsilon \leq \varepsilon_{c2} \rightarrow \sigma_c = 0,85 \cdot f_{cd} \cdot \left[ 1 - \left( 1 - \frac{\varepsilon_c}{\varepsilon_{c2}} \right)^2 \right] \quad (3.11)$$

$$\text{Se } \varepsilon_{c2} < \varepsilon \leq \varepsilon_{cu} \rightarrow \sigma_c = 0,85 \cdot f_{cd}$$

A forma usada pra resolver o problema é semelhante à sugerida por Santos (1994): a tensão no concreto é função da deformação do ponto analisado e a deformação no ponto analisado é função de sua altura em relação à linha neutra. Logo pode-se considerar as tensões

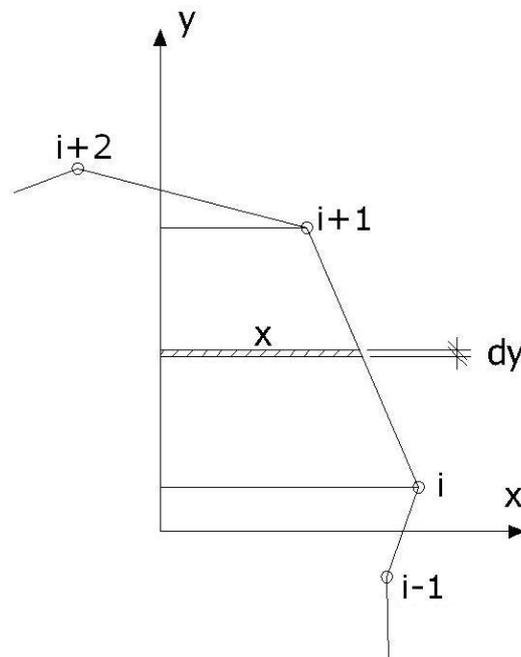
no concreto uma função de  $y$ , que é sempre perpendicular a linha neutra. Expondo matematicamente, tem-se as relações seguintes, levando a relação (3.12):

$$\begin{aligned}\sigma_c &= f(\varepsilon) \\ \varepsilon &= g(y) \\ \sigma_c &= f(g(y))\end{aligned}\quad (3.12)$$

No polígono, as coordenadas  $(x_n, y_n)$  do ponto  $n$  são conhecidas previamente. Assim, a área será derivada em relação a  $y$ , e  $x$  será uma função de  $y$  também, assim como consta em (3.13). Para apresentar de forma mais intuitiva, apresenta-se o raciocínio na Figura 21.

$$\begin{aligned}x &= h(y) \\ A &= \int h(y)dy \\ dA &= h(y)dy \\ dA &= x dy\end{aligned}\quad (3.13)$$

Figura 21 - Faixa infinitesimal  $dy$  na área de concreto.



Conhecendo a expressão para definir a área, e as funções que relacionam  $x$  e  $y$ , deve-se então equacionar as relações entre as tensões, e os momentos em torno de cada eixo ( $M_{xx}$  relativo ao eixo  $x$ , com origem no centro de gravidade da seção, e  $M_{yy}$  relativo ao eixo  $y$ , com origem no centro de gravidade da seção, também).

Assim, são reescritas as expressões seguindo os conceitos apresentados. Elas devem ser calculadas várias vezes por interação: primeiramente até igualar as forças normais na

seção, isto é, procurar a força normal esperada pelo usuário. Então, calculam-se os momentos resultantes dessa atribuição de deformações, obtendo-se (3.14), (3.15) e (3.16).

$$N_c = \int_{A_{cc}} \sigma_{cd} \cdot dA$$

$$N_c = \sum_{i=1}^n \int_{y_i}^{y_{i+1}} \sigma_{cd} \cdot x \, dy \quad (3.14)$$

$$M_{c,xx} = \int_{A_{cc}} \sigma_{cd} \cdot y \cdot dA$$

$$M_{c,xx} = \sum_{i=1}^n \int_{y_i}^{y_{i+1}} \sigma_{cd} \cdot y \cdot x \, dy \quad (3.15)$$

$$M_{c,yy} = \int_{A_{cc}} \sigma_{cd} \cdot x \cdot dA$$

$$M_{c,yy} = \sum_{i=1}^n \int_{y_i}^{y_{i+1}} \sigma_{cd} \cdot \frac{x}{2} \cdot x \, dy = \sum_{i=1}^n \int_{y_i}^{y_{i+1}} \sigma_{cd} \cdot \frac{x^2}{2} \, dy \quad (3.16)$$

Na integral para calcular o  $M_{c,yy}$  há a divisão por 2 pois a força é constante na faixa  $dy$  analisada, e como o centro geométrico do retângulo é na metade de sua base, usou-se  $x$  dividido por 2 como o centro de aplicação da força normal.

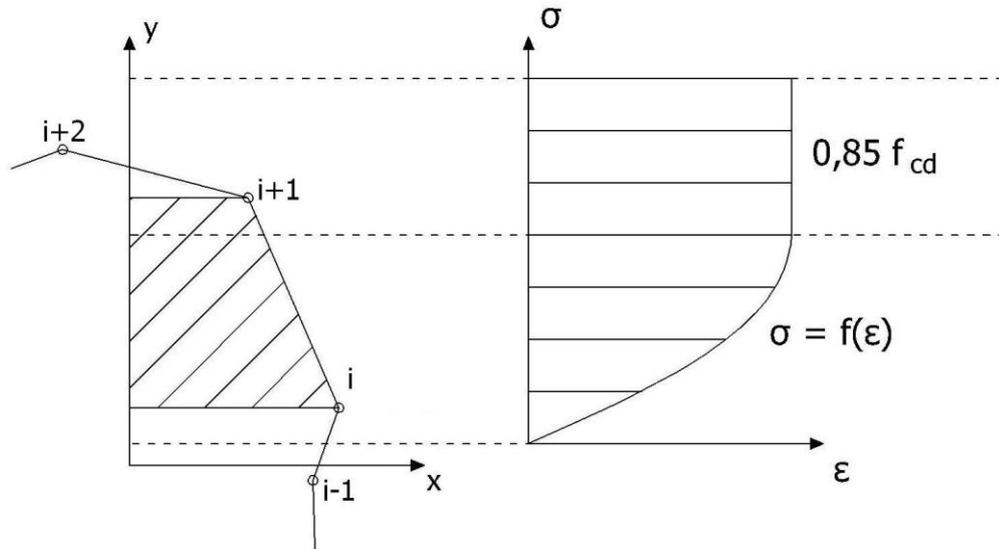
Porém, a função apresenta uma descontinuidade nos pontos  $\varepsilon=0$ ,  $\varepsilon= \varepsilon_{c2}$ ,  $\varepsilon= \varepsilon_{cu}$ , portanto, as funções serão subdivididas em diferentes intervalos, que possuem diferentes equações para a tensão.

Na avaliação dos pontos para distinguir qual equação deve ser utilizada, avaliam-se os pontos  $y$ , comparados aos limites das funções. Os limites são os citados e devem ser vistos como descontinuidades da função tensão em relação à  $\varepsilon$ .

Primeiramente, calculam-se as alturas em que ocorrem as descontinuidades citadas. Então, a cada ponto do polígono em que se calcula a força e momentos do concreto, compara-se os pontos com tais alturas notáveis. Assim, sabe-se qual expressão pra tensão será usada.

No caso dos pontos serem em regiões diferentes, a expressão da integração é considerada usando um somatório de diferentes expressões, usando um ponto intermediário. Assim, substitui-se os intervalos de integração iniciais que são de  $y_n$  a  $y_{n+1}$  por de  $n$  a  $y_b$  somado a outra integral de  $y_b$  até  $y_{n+1}$ , cada um com uma equação diferente. A Figura 22 e a Figura 23 mostram o raciocínio e equacionamento adotado para as alturas e limites de integração.

Figura 22 – Área de concreto sujeita a equações de tensão diferentes.



A exemplificação seguinte apresenta a equação para força axial, porém para momentos segue-se raciocínio análogo, isto é, usa-se um ponto intermediário para o cálculo da integral.

$$N_{c,i} = \int_{y_i}^{y_{i+1}} \sigma_{cd} \cdot x \, dy = \int_{y_i}^{y_b} \sigma_{cd} \cdot x \, dy + \int_{y_b}^{y_{i+1}} \sigma_{cd} \cdot x \, dy$$

Evidentemente, as expressões para o cálculo nos diversos limites diferem entre si.

As alturas notáveis citadas são mostradas pela Figura 23, e são:

$$\varepsilon < 0 \rightarrow y < y_{\max} - x; \sigma_{cd} = \sigma_{cd,0} = 0$$

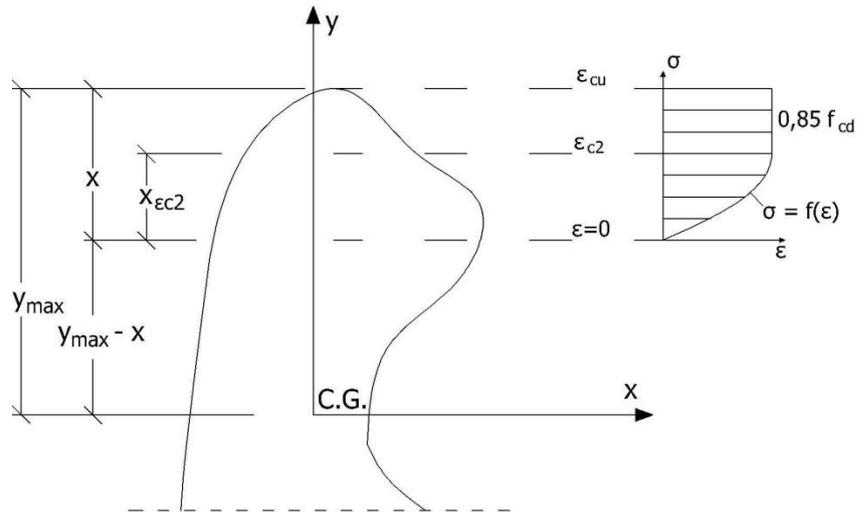
$$0 < \varepsilon < \varepsilon_{c2} \rightarrow y_{\max} - x < y < y_{\max} - x + x_{\varepsilon c2}; \sigma_{cd} = \sigma_{cd,1} = 0,85 \cdot f_{cd} \cdot \left[ 1 - \left( 1 - \frac{\varepsilon_c}{\varepsilon_{c2}} \right)^2 \right]$$

$$\varepsilon_{c2} < \varepsilon \rightarrow y_{\max} - x + x_{\varepsilon c2} < y; \sigma_{cd} = \sigma_{cd,2} = 0,85 \cdot f_{cd}$$

Destaca-se que a distinção entre as diferentes equações para a tensão no concreto é realizada nessa seção, tornando-a mais difícil de entender, porém evita a criação de outra seção com as mesmas avaliações lógicas.

Para facilitar a visualização dos limites e auxiliar no raciocínio e entendimento, apresenta-se a Figura 23. Ela mostra também os valores aos quais o  $y_i$  e  $y_{i+1}$  serão comparados.

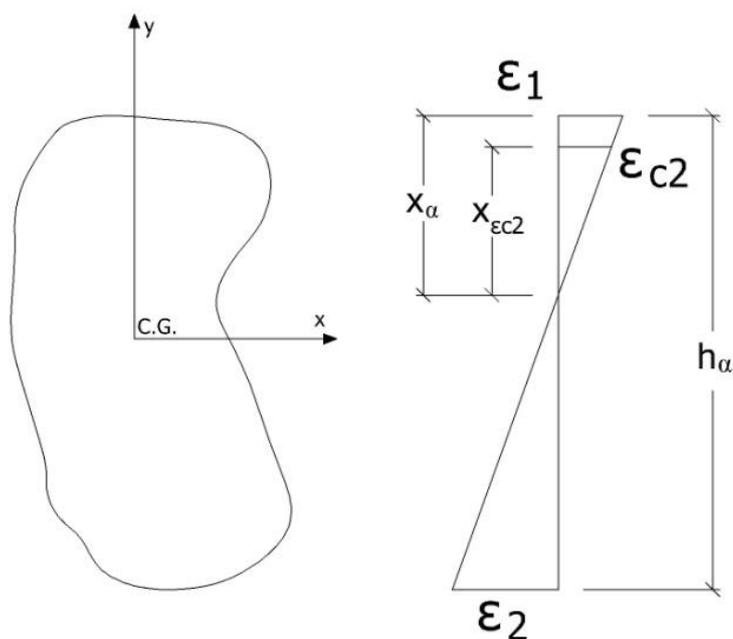
Figura 23 - Variáveis referentes às tensões no concreto.



Na figura, apresentam-se as variáveis  $x_\alpha$  e  $x_{ec2}$ : já mostradas anteriormente, e estas são usadas na rotina para que se possa avaliar qual equação de tensão deve ser usada na área de concreto em questão ( $\sigma_{cd,0}$ ,  $\sigma_{cd,1}$  e  $\sigma_{cd,2}$ ). São também utilizadas para cálculo da função  $\sigma_{cd,1} = f(\epsilon)$ , lembrando que a divisão em 3 intervalos da função  $\sigma_{cd}$  é apresentada e explicada mais detalhadamente posteriormente nesta seção.

Analisando a Figura 24, calcula-se o valor de  $x_\alpha$  comparando os valores de  $x_\alpha$  e  $\epsilon_1$  com os valores de  $h_\alpha$  e  $\epsilon_1 - \epsilon_2$ . Também pode-se calcular o valor de  $x_{ec2}$  comparando os valores de  $x_{ec2}$  e  $\epsilon_{c2}$  com os valores de  $h_\alpha$  e  $\epsilon_1 - \epsilon_2$ .

Figura 24 – Obtenção do valor de  $x_\alpha$  e  $x_{ec2}$



Portanto, obtém-se a equações (3.15) e (3.16).

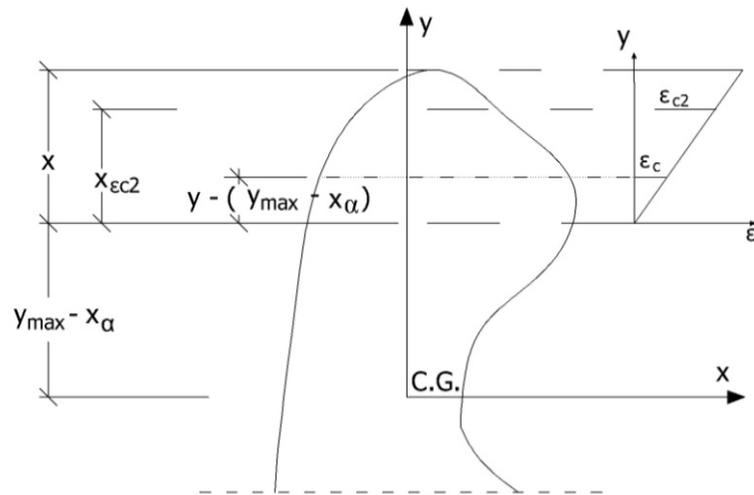
$$x_{\alpha} = \frac{h_{\alpha}}{\varepsilon_1 - \varepsilon_2} \cdot \varepsilon_1 \quad (3.17)$$

$$x_{\varepsilon c2} = \frac{h_{\alpha}}{\varepsilon_1 - \varepsilon_2} \cdot \varepsilon_{c2} \quad (3.18)$$

Além disso, como foi citado, deve-se escrever a função para a tensão em termos de  $y$ , isto é, retoma-se o raciocínio da função composta mostrada na equação (3.12), pra isso, deve-se substituir quaisquer valores de  $\varepsilon_c$  pela função que relaciona  $\varepsilon_c$  e  $y$ .

Para isso, analisando a Figura 25, compara-se  $y - (y_{max} - x_{\alpha})$  e  $\varepsilon_c$  com os valores de  $x_{\varepsilon c2}$  e  $\varepsilon_{c2}$ .

Figura 25 - Relação entre posição vertical e deformação das fibras de concreto.



E por semelhança de triângulos, obtém-se a relação seguinte (3.19), que é válida pelo menos de  $y_{max} - x_{\alpha}$  até  $(y_{max} - x_{\alpha} + x_{\varepsilon c2})$ :

$$\frac{\varepsilon_c}{\varepsilon_{c2}} = \frac{(y - (y_{max} - x_{\alpha}))}{x_{\varepsilon c2}} \quad (3.19)$$

Finalmente, pode-se escrever  $\sigma_{cd,1} = f(g(y))$ , isto é, a equação da tensão em termos de  $y$ , permitindo que se resolva as integrais citadas anteriormente, substituindo (3.19) em (3.11), e obtendo-se (3.20).

$$\sigma_{cd,1} = 0,85 \cdot f_{cd} \cdot \left[ 1 - \left( 1 - \frac{(y - (y_{max} - x_{\alpha}))}{x_{\varepsilon c2}} \right)^2 \right] \quad (3.20)$$

Com tal função determinada, escreve-se as rotinas de comparação, para que o programa consiga distinguir as regiões de diferentes funções no concreto (ver função (3.11)). Assim, tem-se as condições:

Se  $y_n < y_{n+1}$ :

- Se  $y_n < y_{\max} - x$ : (O primeiro ponto está numa área tracionada)

- Se  $y_{n+1} \leq y_{\max} - x$ ; (O segundo ponto está numa área tracionada):

A região em questão não contribui com resistência, força e momento oriundo do concreto na região equivale a zero.

$$N_c = \int_{y_n}^{y_{n+1}} \sigma_{cd,0} \cdot x \, dy = 0$$

- Se  $y_{\max} - x < y_{n+1} \leq y_{\max} - x + x_{ec2}$ ; (O segundo ponto está na primeira área comprimida, em que a tensão varia conforme a deformação):

Deve-se resolver as integrais com os limites de integração, destacando-se que os limites serão usados também no cálculo dos momentos, e não são apresentados pois seria repetição desnecessária:

$$N_c = \int_{y_n}^{y_{n+1}} \sigma_{cd} \cdot x \, dy = \int_{y_n}^{y_b} \sigma_{cd} \cdot x \, dy + \int_{y_b}^{y_{n+1}} \sigma_{cd} \cdot x \, dy$$

$$N_c = \int_{y_n}^{y_{\max-x}} \sigma_{cd,0} \cdot x \, dy + \int_{y_{\max-x}}^{y_{n+1}} \sigma_{cd,1} \cdot x \, dy$$

$$N_c = 0 + \int_{y_{\max-x}}^{y_{n+1}} \sigma_{cd} \cdot x \, dy = \int_{y_{\max-x}}^{y_{n+1}} \sigma_{cd,1} \cdot x \, dy$$

- Se  $y_{\max} - x + x_{ec2} < y_{n+1}$ ; (O segundo ponto está na segunda área comprimida, de tensão é constante em relação a deformação):

$$N_c = \int_{y_n}^{y_{n+1}} \sigma_{cd} \cdot x \, dy = \int_{y_n}^{y_b} \sigma_{cd} \cdot x \, dy + \int_{y_b}^{y_{n+1}} \sigma_{cd} \cdot x \, dy$$

$$N_c = \int_{y_n}^{y_{\max-x}} \sigma_{cd,0} \cdot x \, dy + \int_{y_{\max-x}}^{y_{\max-x+x_{ec2}}} \sigma_{cd,1} \cdot x \, dy + \int_{y_{\max-x+x_{ec2}}}^{y_{n+1}} \sigma_{cd,2} \cdot x \, dy$$

$$N_c = 0 + \int_{y_{\max-x}}^{y_{\max-x+x_{ec2}}} \sigma_{cd,1} \cdot x \, dy + \int_{y_{\max-x+x_{ec2}}}^{y_{n+1}} \sigma_{cd,2} \cdot x \, dy$$

$$N_c = \int_{y_{\max-x}}^{y_{\max-x+x_{ec2}}} \sigma_{cd,1} \cdot x \, dy + \int_{y_{\max-x+x_{ec2}}}^{y_{n+1}} \sigma_{cd,2} \cdot x \, dy$$

• Se  $y_{\max} - x \leq y < y_{\max} - x + x_{ec2}$ : (O primeiro ponto está na primeira área sob compressão, em que a tensão varia conforme a deformação):

- Se  $y_{n+1} \leq y_{\max} - x + x_{ec2}$ ; (O segundo ponto também está na primeira área comprimida, em que a tensão varia conforme a deformação):

Apenas o termo em tal área é utilizado:

$$N_c = \int_{y_n}^{y_{n+1}} \sigma_{cd} \cdot x \, dy = \int_{y_n}^{y_{n+1}} \sigma_{cd,1} \cdot x \, dy$$

- Se  $y_{\max} - x + x_{ec2} < y_{n+1}$ ; (O segundo ponto está na segunda área comprimida, em que a tensão não varia conforme a deformação):

$$N_c = \int_{y_n}^{y_{\max} - x + x_{ec2}} \sigma_{cd,1} \cdot x \, dy + \int_{y_{\max} - x + x_{ec2}}^{y_{n+1}} \sigma_{cd,2} \cdot x \, dy$$

• Se  $y_{\max} - x + x_{ec2} \leq y_n$ : (O primeiro ponto está na segunda área sob compressão, em que a tensão não varia conforme a deformação):

- Só há uma possibilidade: sendo  $y_n < y_{n+1}$ , a única possibilidade é o segundo ponto também estar na segunda camada sob compressão, logo o valor da tensão no concreto é constante, e os limites de integração usados são:

$$N_c = \int_{y_n}^{y_{n+1}} \sigma_{cd,2} \cdot x \, dy$$

**Se  $y_{n+1} < y_n$ :**

• Se  $y_n \leq y_{\max} - x$ : (O primeiro ponto está numa área tracionada)

- A única possibilidade é ( $y_{n+1} < y_{\max} - x$ ), pois  $y_{n+1} < y_n$ ; Portanto, o segundo ponto também está numa área tracionada.

A região em questão não contribui com resistência, força e momento oriundo do concreto na região equivale a zero.

$$N_c = \int_{y_n}^{y_{n+1}} \sigma_{cd,0} \cdot x \, dy = 0$$

• Se  $y_{\max} - x < y_n \leq y_{\max} - x + x_{ec2}$ : (O primeiro ponto está na primeira área sob compressão, em que a tensão varia conforme a deformação):

- Se  $y_{\max} - x < y_{n+1}$ ; (O segundo ponto também está na primeira área comprimida, em que a tensão varia conforme a deformação):

Portanto, apenas o termo para a tensão em tal área é utilizado, com os limites dos 2 pontos apenas:

$$N_c = \int_{y_n}^{y_{n+1}} \sigma_{cd,1} \cdot x \, dy$$

- Se  $y_{n+1} \leq y_{\max} - x$ ; (O segundo ponto está numa área tracionada):

A área de concreto comprimida contribui com resistências, e a área tracionada não. Assim, a área de concreto a se considerar esta na primeira área comprimida, em que a tensão no concreto varia com sua deformação. Os limites de integração usados são:

$$N_c = \int_{y_n}^{y_{\max-x}} \sigma_{cd,1} \cdot x \, dy + \int_{y_{\max-x}}^{y_{n+1}} \sigma_{cd,0} \cdot x \, dy$$

$$N_c = \int_{y_n}^{y_{\max-x}} \sigma_{cd,1} \cdot x \, dy + 0 = \int_{y_n}^{y_{\max-x}} \sigma_{cd,1} \cdot x \, dy$$

• Se  $y_{\max} - x + x_{ec2} < y_n$ : (O primeiro ponto está na segunda área sob compressão, em que a tensão não varia conforme a deformação):

- Caso  $y_{\max} - x + x_{ec2} \leq y_{n+1}$ , o segundo ponto também está na segunda área de tensões de compressão, em que a tensão é constante em relação à deformação. Logo, usa-se os seguintes limites de integração:

$$N_c = \int_{y_n}^{y_{n+1}} \sigma_{cd,2} \cdot x \, dy$$

- Se  $y_{\max} - x \leq y_{n+1} < y_{\max} - x + x_{ec2}$ : O segundo ponto está na primeira área comprimida, em que a tensão varia conforme a deformação):

Portanto:

$$N_c = \int_{y_n}^{y_{n+1}} \sigma_{cd} \cdot x \, dy =$$

$$N_c = \int_{y_n}^{y_{\max-x+x_{ec2}}} \sigma_{cd,2} \cdot x \, dy + \int_{y_{\max-x+x_{ec2}}}^{y_{n+1}} \sigma_{cd,1} \cdot x \, dy$$

- Se  $y_{n+1} < y_{\max} - x$ : O segundo ponto está na área tracionada:

$$N_c = \int_{y_n}^{y_{\max-x+x_{ec2}}} \sigma_{cd,2} \cdot x \, dy + \int_{y_{\max-x+x_{ec2}}}^{y_{\max-x}} \sigma_{cd,1} \cdot x \, dy + \int_{y_{\max-x}}^{y_{n+1}} \sigma_{cd,0} \cdot x \, dy$$

$$N_c = \int_{y_n}^{y_{max-x+x_{ec2}}} \sigma_{cd,2} \cdot x \, dy + \int_{y_{max-x+x_{ec2}}}^{y_{max-x}} \sigma_{cd,1} \cdot x \, dy + 0$$

$$N_c = \int_{y_n}^{y_{max-x+x_{ec2}}} \sigma_{cd,2} \cdot x \, dy + \int_{y_{max-x+x_{ec2}}}^{y_{max-x}} \sigma_{cd,1} \cdot x \, dy$$

Finalmente, com todas as expressões apresentadas, devem ser calcular as integrais: porém, o cálculo de integrais é um procedimento relativamente complexo, em que se observam duas alternativas: as integrais sejam resolvidas com uso de processos numéricos, ou que as expressões sejam simplificadas e possam ser aplicadas em rotinas mais simples, pois as linguagens de programação padrão não calculam integrais automaticamente, tendo em vista sua dificuldade.

A opção seguida por esse trabalho é desenvolver as expressões, considerada menos dispendiosa em relação à capacidade de processamento de dados. Portanto a expressão deve ser simplificada de maneira a torná-la mais simples para posterior uso nos programas.

Anteriormente, foi vista a divisão de  $\sigma_{cd}$  em 3 maneiras diferentes:

- $\sigma_{cd,0}$ : a parte tracionada da seção, que não colabora com resistências, assim como proposto como hipótese inicial: o concreto não apresenta tensões de tração (foram ignoradas).

- $\sigma_{cd,1}$ : já nesse trecho, a tensão varia com a deformação, e a deformação varia conforme a altura do ponto em relação à linha neutra: pontos na linha neutra tem deformação zero, enquanto pontos superiores a esta tem deformações positivas, de módulo maior quanto maior a distância à linha neutra. Assim, para que o cálculo da integral seja possível, a tensão é escrita em função da altura em relação à linha neutra  $y$ .

- $\sigma_{cd,2}$ : no último trecho, a tensão é constante em relação à deformação, porém, como a seção tem forma variável, a integral deve ser resolvida, a fim de apresentar confiabilidade na teoria. Estima-se que a equação para este último trecho irá sugerir que a força é aplicada no centro geométrico da figura, pois como a tensão é constante, o centro de aplicação corresponde ao centro geométrico.

Calculam-se as expressões a serem usadas por 6 integrais diferentes:  $N_c$ ,  $M_{c,xx}$  e  $M_{c,yy}$  para a tensão  $\sigma_{cd,1}$ , e  $N_c$ ,  $M_{c,xx}$  e  $M_{c,yy}$  para a tensão  $\sigma_{cd,2}$ .

Ressalta-se que a expressão usada para as tensões do concreto é em função de  $y$ , como já apresentado anteriormente, e desenvolvido até obter-se a expressão (3.20).

Finalmente, tem-se as expressões usadas para o cálculo das forças normais e dos momentos. Porém, devido ao tamanho das expressões, é conveniente usar algumas

simplificações para facilitar o entendimento e tornar as equações de manuseio mais fácil. Foi usado o recurso computacional Wolfram|Alpha em junho de 2014.

Simplificações adotadas:

$$c_1 = \frac{y_1 \cdot x_2 - y_2 \cdot x_1}{y_1 - y_2}$$

$$c_2 = \frac{x_1 - x_2}{y_1 - y_2}$$

$$x_y = c_1 + c_2 \cdot y$$

$$g = y_{max} - x_\alpha$$

Essas constantes,  $c_1$  e  $c_2$ , são usadas a cada ponto, e calculam a equação da reta, visto que são retas que unem a poligonal. Foram usados para calcular o  $x_y$ : não a altura da linha neutra, mas  $x_y = f(y)$ . Além disso, para não aumentar o tamanho do programa, a potência foi substituída pelo exponencial do logaritmo neperiano da base e o produto deste com o expoente, isto é, substituindo  $a^b$  por  $e^{\ln(a) \cdot b}$ .

Nos limites em que  $\sigma_{cd} = \sigma_{si,1}$ , tem-se as seguintes expressões:

-  $N_{cc,1}$ :

$$N_{cd,1} = \int 0,85 \cdot f_{cd} \cdot \left[ 1 - \left( 1 - \frac{(y - (y_{max} - x_\alpha))}{x_{\epsilon c 2}} \right)^n \right] \cdot x_y \cdot dy =$$

$$N_{cd,1} = -0,85 * f_{cd} \cdot \dots$$

$$\left( - \frac{x_{\epsilon c 2} \cdot \left( \frac{g + x_{\epsilon c 2} - y}{x_{\epsilon c 2}} \right)^{n+1} \cdot (c \cdot (n+2) + d \cdot (g + x_{\epsilon c 2} + n \cdot y + y))}{(n+1) \cdot (n+2)} - \dots \right.$$

$$\left. c \cdot y - \frac{d \cdot y^2}{2} \right) + Constante \quad (3.21)$$

-  $M_{cd,1,xx}$ :

$$M_{cd,1,xx} = \int 0,85 \cdot f_{cd} \cdot \left[ 1 - \left( 1 - \frac{(y - (y_{max} - x))}{x_{\epsilon c 2}} \right)^2 \right] \cdot x \cdot y \cdot dy =$$

$$\frac{0,85 \cdot f_{cd}}{6 \cdot (1+n) \cdot (2+n) \cdot (3+n)} \cdot \dots$$

$$\left( 3 \cdot c \cdot \left( (1+n) \cdot (2+n) \cdot (3+n) \cdot y^2 + 2 \cdot x_{\epsilon c 2} \cdot \left( \frac{g + x_{\epsilon c 2} - y}{x_{\epsilon c 2}} \right)^{n+1} \dots \right. \right.$$

$$\left. \left( (g + x_{\epsilon c 2}) \cdot (3+n) + (3 + 4 \cdot n + n^2) \cdot y \right) \right) + \dots$$

$$\begin{aligned}
& 2 \cdot d \cdot \left( (1+n) \cdot (2+n) \cdot (3+n) \cdot y^3 + 3 \cdot x_{\varepsilon c,2} \cdot \left( \frac{g + x_{\varepsilon c,2} - y}{x_{\varepsilon c,2}} \right)^{n+1} \cdot \dots \right. \\
& \left. \left( 2 \cdot g^2 + 2 \cdot x_{\varepsilon c,2}^2 + 2 \cdot x_{\varepsilon c,2} \cdot (1+n) \cdot y + (2+3 \cdot n + n^2) \cdot y^2 + 2 \cdot g \cdot (2 \cdot x_{\varepsilon c,2} + (1+n) \cdot y) \right) \right) \\
& + \text{Constante} \tag{3.22}
\end{aligned}$$

-  $M_{cd,1,yy}$ :

$$\begin{aligned}
M_{cd,1,yy} &= \int 0,85 \cdot f_{cd} \cdot \left[ 1 - \left( 1 - \frac{(y - (y_{max} - x))}{x_{\varepsilon c,2}} \right)^2 \right] \cdot \frac{x^2}{2} \cdot dy = \\
& \frac{0,85 \cdot f_{cd}}{6 \cdot (1+n) \cdot (2+n) \cdot (3+n)} \cdot \dots \\
& \left( 3 \cdot c^2 \cdot \left( x_{\varepsilon c,2} \cdot (6 + 5 \cdot n + n^2) \cdot \left( \frac{g + x_{\varepsilon c,2} - y}{x_{\varepsilon c,2}} \right)^{n+1} + (1+n) \cdot (2+n) \cdot (3+n) \cdot y \right) \dots \right. \\
& + 3 \cdot c \cdot d \cdot \left( (1+n) \cdot (2+n) \cdot (3+n) \cdot y^2 + 2 \cdot x_{\varepsilon c,2} \cdot \left( \frac{g + x_{\varepsilon c,2} - y}{x_{\varepsilon c,2}} \right)^{n+1} \cdot \dots \right. \\
& \left. \left( (g + x_{\varepsilon c,2}) \cdot (3+n) + (3 + 4 \cdot n + n^2) \cdot y \right) \right) \dots \\
& + d^2 \left( (1+n) \cdot (2+n) \cdot (3+n) \cdot y^3 + 3 \cdot x_{\varepsilon c,2} \cdot \left( \frac{g + x_{\varepsilon c,2} - y}{x_{\varepsilon c,2}} \right)^{n+1} \right. \\
& \left. \left. \cdot \left( 2 \cdot g^2 + 2 \cdot x_{\varepsilon c,2}^2 + 2 \cdot x_{\varepsilon c,2} \cdot (1+n) \cdot y + (2+3 \cdot n + n^2) \cdot y^2 + 2 \cdot g \cdot (2 \cdot x_{\varepsilon c,2} + (1+n) \cdot y) \right) \right) \right) \\
& + \text{Constante} \tag{3.23}
\end{aligned}$$

Se a tensão for dada por  $\sigma_{si,2}$ :

-  $N_{cc,2}$ :

$$\begin{aligned}
N_{cd,2} &= \int 0,85 \cdot f_{cd} \cdot x_y \cdot dy = \\
N_{cd,2} &= 0,85 f_{cd} \cdot \left( cy + d \frac{y^2}{2} \right) + \text{Constante} \tag{3.24}
\end{aligned}$$

-  $M_{cd,2,xx}$ :

$$\begin{aligned}
M_{cd,2,xx} &= \int 0,85 \cdot f_{cd} \cdot x \cdot y \cdot dy = \\
&= \frac{0,85 f_{cd} \cdot y^2 \cdot (3 \cdot (y_1 \cdot x_2 - y_2 \cdot x_1) + 2 \cdot (x_2 - x_1) \cdot y)}{6 \cdot (y_1 - y_2)} + \text{Constante} \tag{3.25}
\end{aligned}$$

$$\begin{aligned}
 & - M_{cd,2,yy}: \\
 M_{cd,2,yy} &= \int 0,85 \cdot f_{cd} \cdot \frac{x^2}{2} \cdot dy = \\
 & 0,85 \cdot f_{cd} \cdot \frac{c^2 \cdot y + c \cdot d \cdot y^2 + d^2 \cdot \frac{y^3}{3}}{2} + Constante \quad (3.26)
 \end{aligned}$$

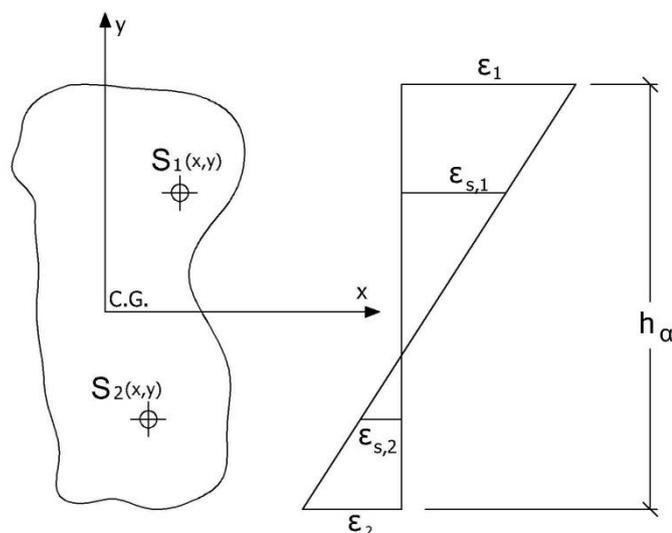
### 3.3 CÁLCULO DA FORÇA E MOMENTOS DA ARMADURA PASSIVA

Enquanto que o apresentado para o concreto é devido à sua distribuição numa área poligonal qualquer, a área de aço será considerada praticamente pontualmente. Portanto, a solução é muito mais fácil: cada barra de aço tem área e posição pré-estabelecidos, e os inúmeros diagramas de deformação serão fixados para cada situação, impondo diversas situações de esforços.

Para cada situação destas, haverá uma resistência axial, e momentos resistidos em cada direção  $u$  e  $v$ , giradas num ângulo  $a$  em relação às coordenadas  $x$  e  $y$ .

A força pontual será calculada usando também as considerações já pré-estabelecido para o aço, que nesta análise será considerado um material elástico até a parte dúctil. A Figura 26 apresenta numa situação exemplo o cálculo que será realizado, e ajuda a visualizar o problema.

Figura 26 - Exemplo de situação de cálculo.



Deduz-se a equação para os valores de deformações de cada barra, para que a rotina seja escrita, apresenta a seguir por (3.27). Além disso, o valor de  $F_{s,i}$  de cada barra é calculado

a partir da definição de tensão, que é função da deformação, e equacionando, chega-se a (3.28):

$$\varepsilon_{s,i} = \varepsilon_1 - \frac{h_{si} (\varepsilon_1 - \varepsilon_2)}{h_\alpha} \quad (3.27)$$

$$F_{s,i} = A_{si} \cdot \sigma_{si}(\varepsilon_{s,i}) \quad (3.28)$$

A abordagem aqui usada recomenda e usará no programa elaborado uma rotina *function* para relacionar deformações e tensões.

Sabendo-se que a força de cada barra será conhecida conforme está é submetida à deformações, pode-se concluir que os momentos em relação ao centro de gravidade dados por uma força num ponto  $x,y$  serão dados pelas equações (3.29) e (3.30):

$$M_{s,i,xx} = F_{s,i} \cdot y_{s,i} \quad (3.29)$$

$$M_{s,i,yy} = F_{s,i} \cdot x_{s,i} \quad (3.30)$$

Portanto, para considerar todas as barras, sendo o número destas  $n$  apenas para termo de equacionamento, tem-se (3.31), (3.32) e (3.33).

$$N_s = \sum_{i=1}^n F_{s,i} \quad (3.31)$$

$$M_{s,xx} = \sum_{i=1}^n F_{s,i} \cdot y_{s,i} \quad (3.32)$$

$$M_{s,yy} = \sum_{i=1}^n F_{s,i} \cdot x_{s,i} \quad (3.33)$$

### 3.4 CÁLCULO DA FORÇA E MOMENTOS DA ARMADURA ATIVA

Além da armadura passiva, adiciona-se uma maneira de considerar eventuais efeitos de protensão na peça. As forma de considerar as tensões se assemelha em muitos aspectos à consideração da armadura passiva, na seção imediatamente anterior 3.4: isto é, a força é considerada pontual, no centro de gravidade da armadura, e deste principio advém todas as equações utilizadas.

As duas principais diferenças são: primeiramente a consideração de uma relação tensão-deformação diferente, que segue a conformação da Figura 11 e as equações (2.9) e (2.10). A segunda diferença é o cálculo da deformação a ser utilizada: a armadura ativa apresenta tensões de tração, forçando tensões de compressão no elemento estrutural: essa tensão é avaliada com base no  $\varepsilon_p$ , isto é, a deformação de pré-alongamento da peça, que deve ser fornecido pelo usuário.

Outra diferença diz respeito à relação entre o diâmetro e área da armadura ativa: que não acompanham a equação tradicional de área de circunferências, pois as cordoalhas possuem espaços vazios, porém o diâmetro da armadura ativa apenas é importante para o cálculo da abertura de fissuras. Por isso, é recomendável que haja um campo para que o usuário possa fornecer o diâmetro, caso deseje uma verificação do estado-limite de abertura de fissuras.

Então, quando na análise da armadura ativa, utiliza-se a equação (3.27) para calcular a deformação causada na armadura pela seção  $\varepsilon_s$ . Então, soma-se a deformação de pré-alongamento, obtendo-se assim o  $\varepsilon_t$ , que será aplicado nas equações (2.9) ou (2.10) para obtenção da tensão da armadura.

Por fim, usa-se: a equação (3.28) para calcular a força axial; e as equações (3.29) e (3.30) para calcular os momentos causados pela força resultante da armadura. Expandindo o cálculo para todas as armaduras, tem-se (3.31), (3.32) e (3.33).

### 3.5 CÁLCULOS FINAIS DA ETAPA

Finalmente, após a força normal ter sido alcançada, calculam-se os momentos  $M_{uu}$  e  $M_{vv}$ , esforços resistentes oriundos do concreto e do aço, os momentos resistentes da conformação de deformações é dada pela soma dos 2.

$$M_{uu} = M_{c,uu} + M_{s,uu}$$

$$M_{vv} = M_{c,vv} + M_{s,vv}$$

Os eixos  $u$  e  $v$  nada mais é que a notação usada para o eixo  $x$  e  $y$  após serem girados por um ângulo  $\alpha$ . Para retornar os momentos calculados, que estão apresentados em torno do eixo  $u$  e  $v$  aos eixos  $x$  e  $y$  usa-se a expressão indicada por Santos (1996), (3.34) e (3.35):

$$M_{xx} = M_{uu} \cdot \cos \alpha + M_{vv} \cdot \sen \alpha \quad (3.34)$$

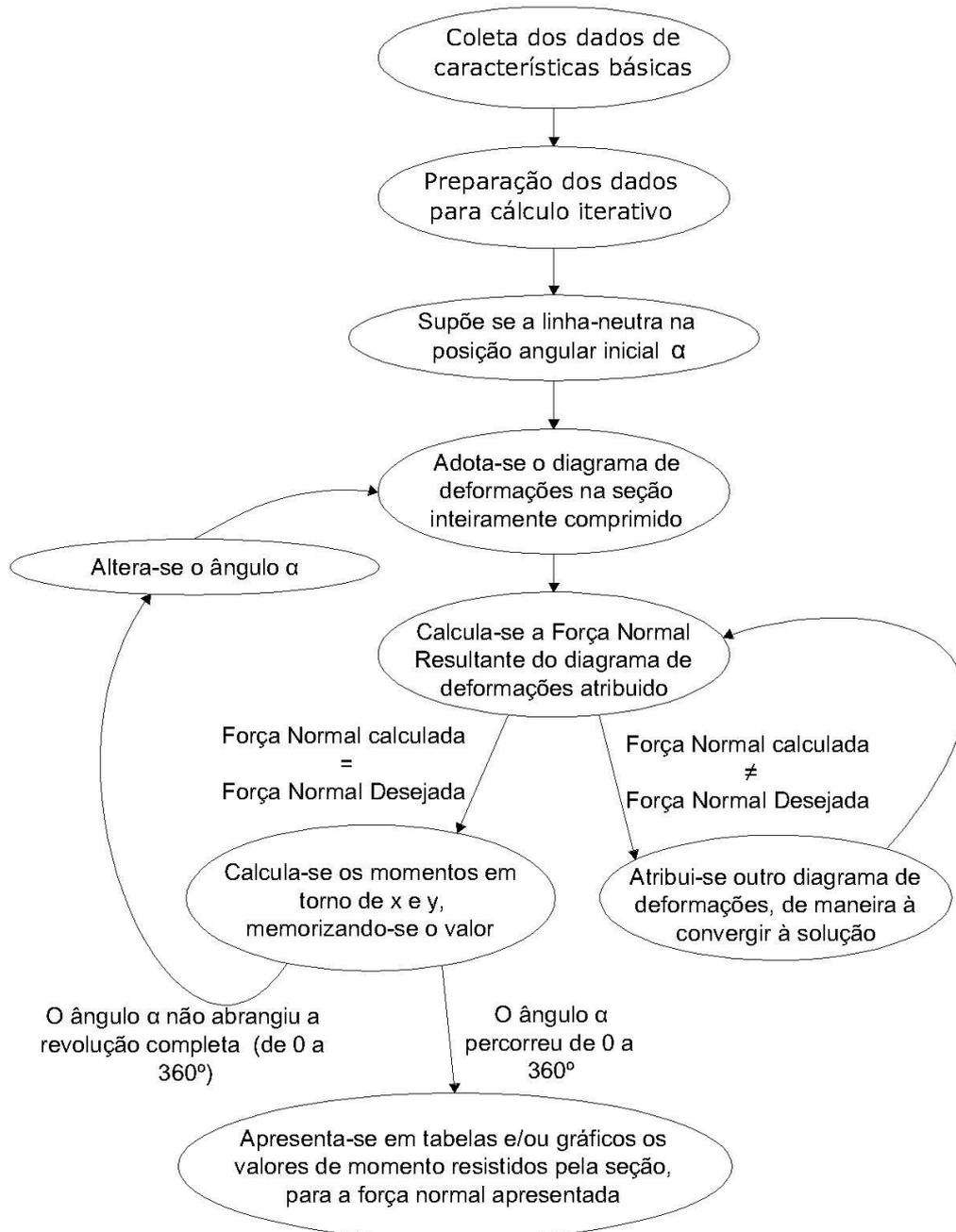
$$M_{yy} = -M_{uu} \cdot \sen \alpha + M_{vv} \cdot \cos \alpha \quad (3.35)$$

Os resultados estão prontos para serem apresentados ao usuário, podendo ser armazenados, conforme a conveniência do usuário também. Alerta-se que, caso a seção tenha sido girada por um ângulo  $\alpha$ , deverá ser girada de volta por um ângulo  $(-\alpha)$ . E o processo será repetido até que a análise tenha sido feita na circunferência inteira, isto é, abrangendo pelo menos  $360^\circ$ , lembrando que no sistema utilizado,  $0^\circ = 360^\circ$ .

### 3.6 ESTADO-LIMITE ÚLTIMO

Tendo em vista as equações propostas por Fusco (1981) e Marino (1979), e os passos descritos por Santos (1994), escreveu-se o algoritmo apresentado na Figura 27. Tal algoritmo não detalha muitas etapas importantes, mas dá uma ideia geral do raciocínio usado no programa.

Figura 27 - Algoritmos para resolução da FCO.



Na Figura 27 também se indica que se atribui outro diagrama de deformações, de maneira a convergir a solução. O método de convergência é o método numérico apresentado na seção 3.8, e explica-se como aplicá-lo no estado-limite último na seção 3.8.1.

### 3.7 ESTADO-LIMITE DE SERVIÇO

Orienta-se nesta seção sobre as rotinas para obtenção do gráfico momento-curvatura e verificações do estado limite de serviço relativas à protensão: abertura de fissuras (ELS-W), formação de fissuras (ELS-F) e descompressão (ELS-D).

Tais rotinas não foram condensadas em apenas um arquivo importável, como foram feitas com outras rotinas, como descreve especificamente o capítulo 0, devido às rotinas necessitarem interagir com objetos do programa (mostrar a resposta num campo de texto, ou montar um gráfico com uma dupla de valores, por exemplo), fazendo com que fosse especialmente difícil resumi-las sem prejudicar parte das rotinas ou do próprio programa.

Portanto, na intenção de realizar alterações ou ainda reutilizar essas rotinas não importáveis, sugere-se a leitura deste capítulo e a análise do algoritmo do programa: tendo em vista a complexidade que vêm com a leitura de algoritmos, acredita-se que esta seção e suas explicações possam facilitá-la.

#### 3.7.1 Gráfico Momento-Curvatura

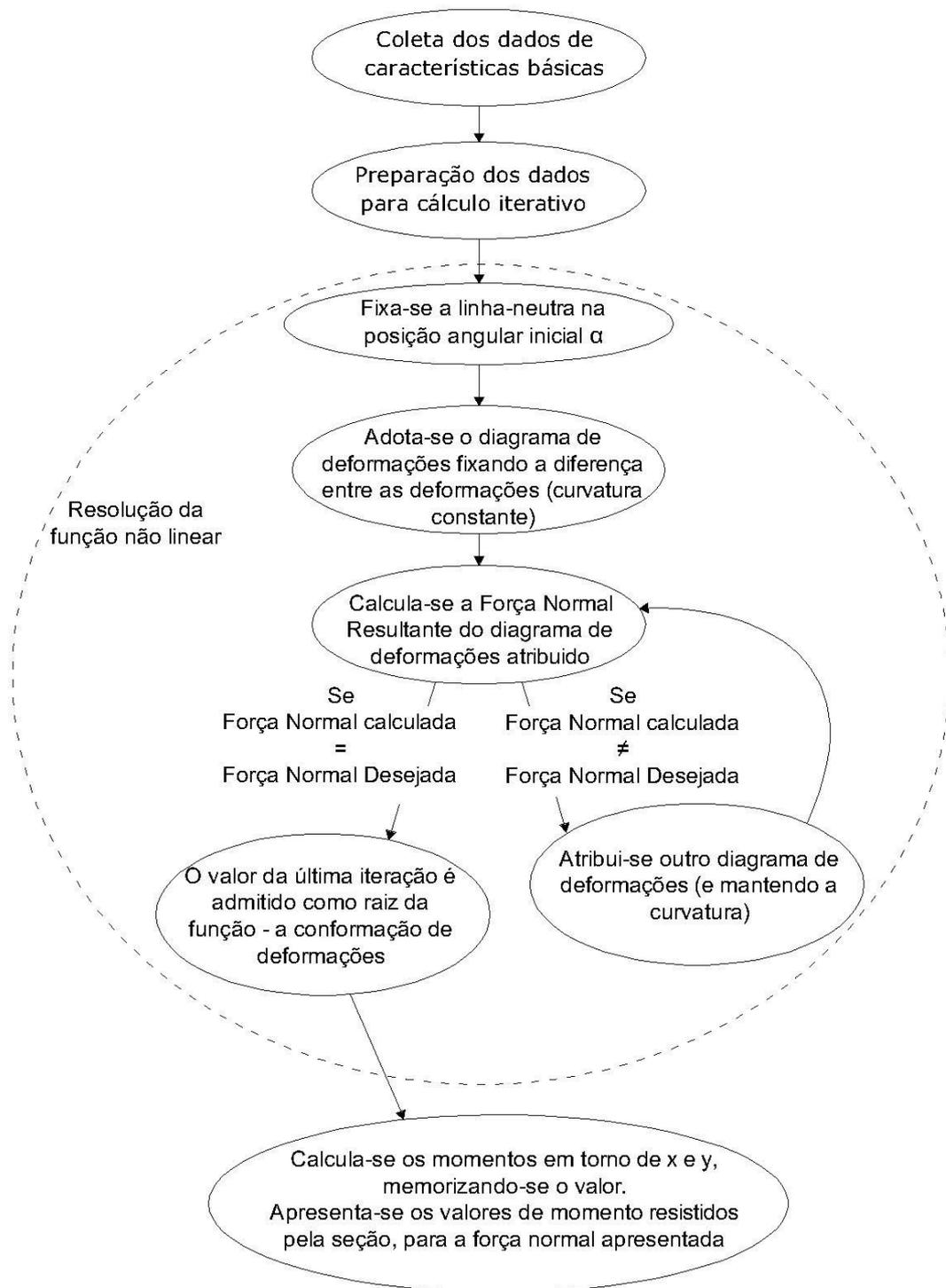
Analogamente ao algoritmo apresentado na seção 3.6 para estado-limite último, apresenta-se o algoritmo usado para analisar o estado-limite de serviço, Figura 28. O produto final de tal algoritmo é o gráfico momento-curvatura.

Seguindo o raciocínio usado em estado-limite último, recomendações de Santos (1994) e Fusco (1981) segue o algoritmo referente ao cálculo da relação momento-curvatura:

Variam-se as deformações, de forma a realizar uma varredura das possibilidades à procura da força normal fixada (pois recapitulando, a relação gráfica mostrada na seção 2.6.1 é calculada para uma força axial constante). Para cada conformação de curvatura (crescente, por convenção), há um valor de momento apresentado pela seção, e com tais valores, apresenta-se a relação momento-curvatura,

Segundo Fusco (1981), ao se fixar inicialmente o valor de  $1/r$ , e posteriormente variar o valor de deformação na seção, fixando e calculando as forças resultantes de deformação por deformação obtém-se os valores de carga axial, e momento em duas direções (segundo a convenção do presente trabalho  $M_{xx}$  e  $M_{yy}$ ). O autor indica que ao se definir o valor da curvatura e da deformação, fica definido univocamente o diagrama de deformações. Além disso, a mesma referência também apresenta um algoritmo para obtenção da relação momento-curvatura da seção.

Figura 28 – Algoritmo usado para obtenção do momento-curvatura.



Tal rotina geral foi adaptada ao programa e suas rotinas: a alteração diz respeito aos limites de ruptura: a cada iteração, calcula-se a força axial mínima e máxima que a seção pode apresentar com aquela curvatura, e caso a força axial constante do gráfico às exceda, considera-se que a seção alcança a ruptura, e finaliza-se o gráfico.

Para resolver o problema, deve-se descobrir a conformação de deformações que apresente a força normal desejada, sob outra perspectiva: a diferença entre  $\varepsilon_1$  e  $\varepsilon_2$  é conhecida previamente, pois os valores de curvatura serão atribuídos. Logo, pode-se supor um em função do outro, assim como indicado pelas equações (3.36) ou (3.37), baseando-se em (2.13), reduzindo a um problema de uma variável apenas.

$$\varepsilon_1 = \varepsilon_2 + \frac{1}{r} \cdot d \quad (3.36)$$

$$\varepsilon_2 = \varepsilon_1 - \frac{1}{r} \cdot d \quad (3.37)$$

Além disso, o ângulo escolhido pra obtenção dos valores é escolhido pelo usuário: portanto, seguindo o mesmo principio do problema de estado-limite último, novamente se trata de um problema de uma variável apenas.

A variável aqui utilizada para indicar a diferença de deformação nas bordas é mostrada em (3.38): e como mostrado, representa a diferença entre a deformação no borda superior e na borda inferior.

$$\Delta\varepsilon = \varepsilon_1 - \varepsilon_2 \quad (3.38)$$

Para resolvê-lo, primeiramente consideram-se os limites máximos de esforço axial da seção: força axial máxima e mínima.

No limite mínimo de deformações, será suposto que a ruptura ocorre por ruptura do aço no ponto A (ver Figura 4 ou Figura 5), na deformação  $\varepsilon_2 = \varepsilon_{uk}$  (que tem valor padrão estabelecido como -10‰), e  $\varepsilon_1$  obtido por (3.38).

No limite máximo de deformações, será suposto que a ruptura ocorre por deformações excessivas de compressão no concreto, que poderá ocorrer no ponto B ou C (novamente, seguindo indicado em Figura 4 e Figura 5), e para entender as inequações dos dois parágrafos seguintes, sugere-se a análise gráfica dos pontos B e C e o modo de ruptura:

- Ruptura por compressão no ponto B: ocorre quando  $\varepsilon_1 - \varepsilon_2 \geq \varepsilon_{cu}$ . Retomando a equação (3.38), tem-se  $\Delta\varepsilon \geq \varepsilon_{cu}$  nesta região, a ruptura ocorre em torno do ponto B e as deformações usadas no cálculo da força axial máxima são dadas por (3.39).

$$\varepsilon_1 = \varepsilon_{cu} \quad (3.39)$$

- Ruptura por compressão no ponto C: ocorre quando  $\varepsilon_1 - \varepsilon_2 < \varepsilon_{cu}$ . Retomando a equação (3.38), pode-se verificar que é válido  $\Delta\varepsilon < \varepsilon_{cu}$ . Nesta região, a ruptura ocorre em torno do ponto C e as deformações usadas no cálculo da força axial máxima são mostradas como mostrado em (3.40), obtidas a partir de (3.27) e simplificando matematicamente.

$$d_{\varepsilon c2} = \frac{\varepsilon_{cu} - \varepsilon_{c2}}{\varepsilon_{cu}} \cdot h$$

$$\varepsilon_1 = \varepsilon_{c2} + \frac{d_{\varepsilon c2} \cdot (\varepsilon_1 - \varepsilon_2)}{h} = \varepsilon_{c2} + \frac{d_{\varepsilon c2} \cdot \Delta\varepsilon}{h} \quad (3.40)$$

Assim, ao se calcular as forças axiais mínima e máxima, pode-se continuar o cálculo, buscando a conformação de deformação que corresponde à força axial desejada, e registrando o momento de tal conformação.

Aplica-se tal método até que a força axial fornecida pelo usuário  $N_u$  exceda o valor mínimo ou máximo, quando isso ocorrer o gráfico esta terminado. Nota-se, que ao aplicar o método das secantes, podem-se usar os valores máximos e mínimos como pontos iniciais do processo iterativo, o que fica mais claro na seção 3.8.2.

### 3.7.2 Verificações dos estados-limite de serviço

Embora na seção 2.6 os estados-limites de serviço são tratados separadamente, neste tópico é conveniente abordar todos simultaneamente, devido à terem a mesma origem e terem resoluções que envolvem as mesmas variáveis: em todos os casos, deseja-se verificar seção conhecida segundo solicitações pré-estabelecidas, logo, todos poderiam ser resolvidos em paralelo mesmo se suas resoluções fossem estritamente diferentes.

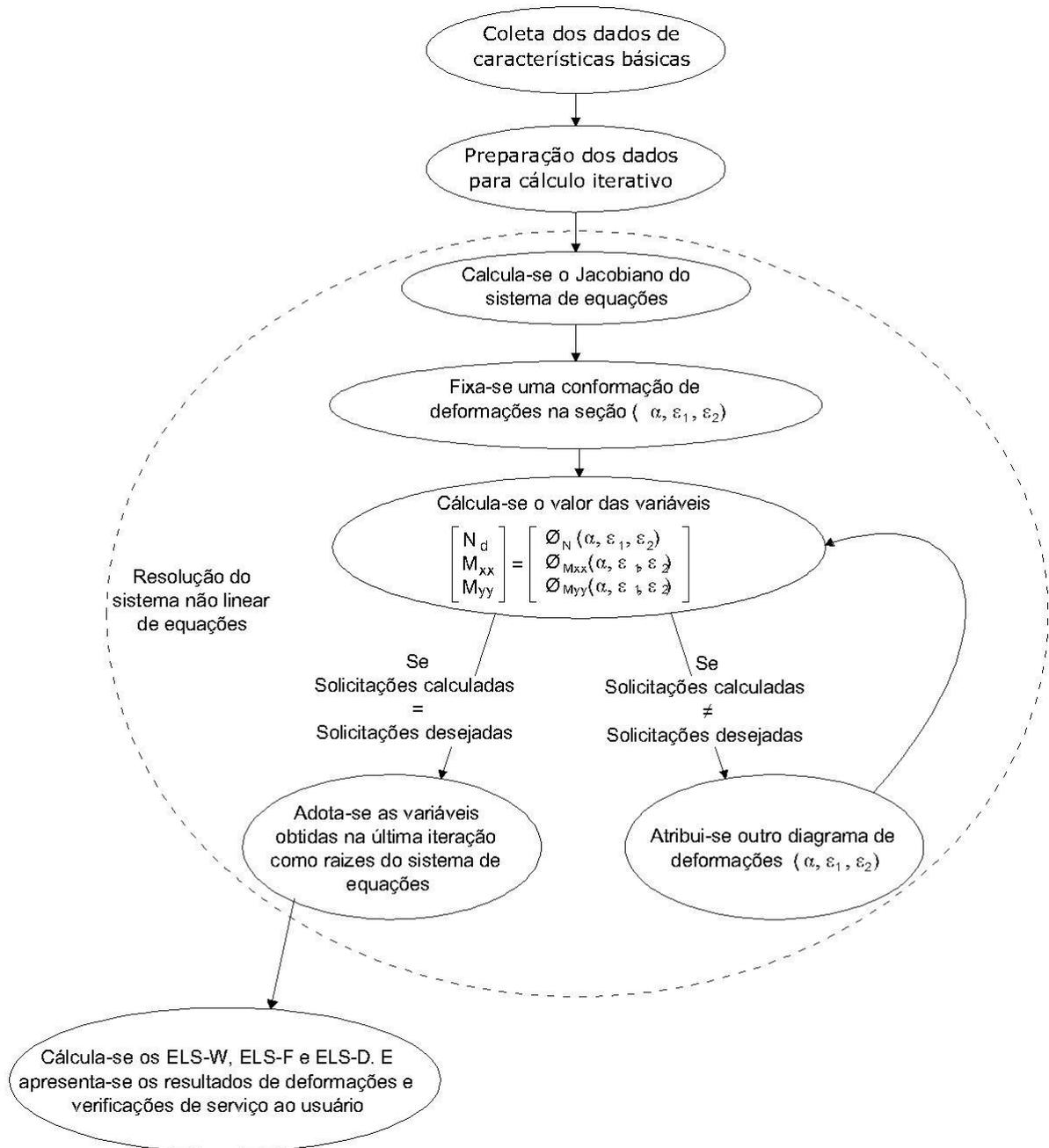
Para resolver esses casos, parte-se da suposição de uma seção, sua armadura passiva e armadura ativa, e de materiais conhecidos e também um conjunto de solicitações normais aplicados a ela: força axial e momento fletor (em qualquer direção). Também se admite que a solução seja unívoca, baseando-se no indicado segundo Fusco (1981), e para obtenção dos resultados, apresentam-se a seguinte rotina:

Sendo as variáveis  $N_u$ ,  $M_{xx,u}$  e  $M_{yy,u}$  fornecidas pelo usuário, sabe-se que as rotinas elaboradas calculam a partir dos valores do ângulo  $\alpha$ ,  $\varepsilon_1$  e  $\varepsilon_2$ , e fornecem os valores situacionais de força axial  $N$ , e momento nas direções  $x$  e  $y$ , respectivamente  $M_{xx}$  e  $M_{yy}$ .

Escrevendo matematicamente, tem-se (3.41), e a Figura 29 apresenta um algoritmo que resume a resolução do problema:

$$\begin{bmatrix} N \\ M_{xx} \\ M_{yy} \end{bmatrix} = \begin{bmatrix} \Phi_N(\alpha, \varepsilon_1, \varepsilon_2) \\ \Phi_{M_{xx}}(\alpha, \varepsilon_1, \varepsilon_2) \\ \Phi_{M_{yy}}(\alpha, \varepsilon_1, \varepsilon_2) \end{bmatrix} \quad (3.41)$$

Figura 29 – Algoritmo para verificação de estado-limite de serviço.



Obtendo a conformação de  $\alpha$ ,  $\varepsilon_1$  e  $\varepsilon_2$  que representem as solicitações a serem verificadas  $N_u$ ,  $M_{xx,u}$  e  $M_{yy,u}$ , é possível verificar os estados-limites:

- Estado-limite de abertura de fissuras (ELS-W): Com a deformação sendo calculada em cada ponto da seção, calcula-se a tensão em cada armadura usando a equação (3.27), sendo esta a variável a ser usada nas equações de abertura de fissuras, que é a equação (2.22). Um detalhe a ser levado em conta é para a armadura ativa: deve-se calcular sua tensão na deformação obtida, e subtrair a tensão do pré-ologamento, pois a tensão a ser considerada é o

acréscimo de tensão entre o estado-limite de descompressão e do carregamento considerado, como lê-se na norma ABNT NBR 6118:2014, no item 17.3.3.2:

Nos elementos estruturais com protensão,  $\sigma_{si}$  é o acréscimo de tensão, no centro de gravidade da armadura, entre o estado limite de descompressão e o carregamento considerado. Deve ser calculado no estágio II, considerando toda a armadura ativa, inclusive aquela dentro da bainhas.

- Estado-limite de formação de fissuras (ELS-F): Embora a orientação da norma leve em conta uma análise sob outra perspectiva (cálculo do momento de fissuração, seção 17.3.1), propõe-se uma análise de tensões ou deformações, levando em conta o limite de resistência à tração do concreto e/ou seu módulo de elasticidade. No presente trabalho sugere-se a comparação do ponto mais tracionado da seção ( $\epsilon_2$ ) com um valor limite: caso o valor seja excedido, o estado-limite não é atendido, e caso contrário, o estado-limite é atendido. Mais informações sobre este estado-limite constam na seção 2.6.3 deste trabalho, e na norma ABNT NBR 6118:2014, mais especificamente em 3.2.2 e nas seções 13.4.2 e 17.3.4.

- Estado-limite de descompressão (ELS-D): analisa-se o ponto de tração máxima ( $\epsilon_2$ ). Se o ponto apresentar tração, o estado-limite não é atendido. Caso a tensão for numa ou de compressão, o estado-limite é atendido.

### 3.8 MÉTODO DE NEWTON E MÉTODO DA SECANTE

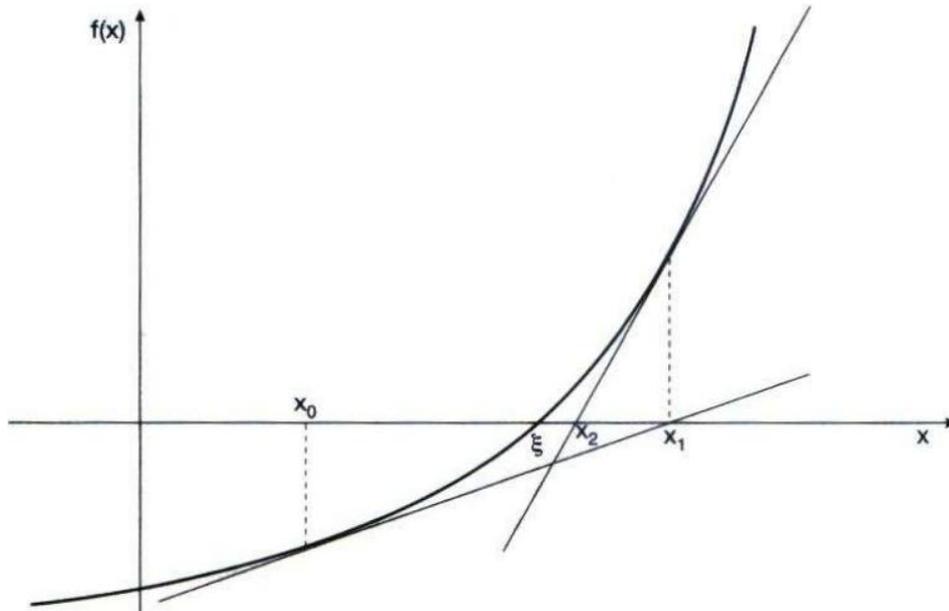
As rotinas apresentadas baseiam-se em procurar os diagramas de conformação que fornecem os valores desejados de força normal. Assim sendo, torna-se viável o uso de métodos de cálculo numérico para melhorar a velocidade de convergência, isto é, aplicar um método que reduza o número de iterações necessárias, melhorando a velocidade de processamento do programa.

Um dos métodos mais populares, como descrevem as bibliografias Ruggiero e Lopes (2006) e Franco (2006), e com ótima velocidade de convergência é o Método de Newton (ou Newton-Raphson). Este método visa obtenção de raízes de uma equação, e para isso considera-se o valor da função analisada em um certo ponto ( $x_k, f(x_k)$ ). Então, traça-se a reta tangente a este ponto e encontra-se o zero desta reta, considerando que a reta é uma aproximação da função  $f(x)$  numa vizinhança de  $x_k$ .

Observa-se o comportamento numa função qualquer do método analisando-se a Figura 30, que mostra a interpretação geométrica do método de Newton, cuja equação que apresenta o método é mostrada por (3.42).

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (3.42)$$

Figura 30 – Interpretação Geométrica do Método de Newton



Fonte: Ruggiero e Lopes (2006).

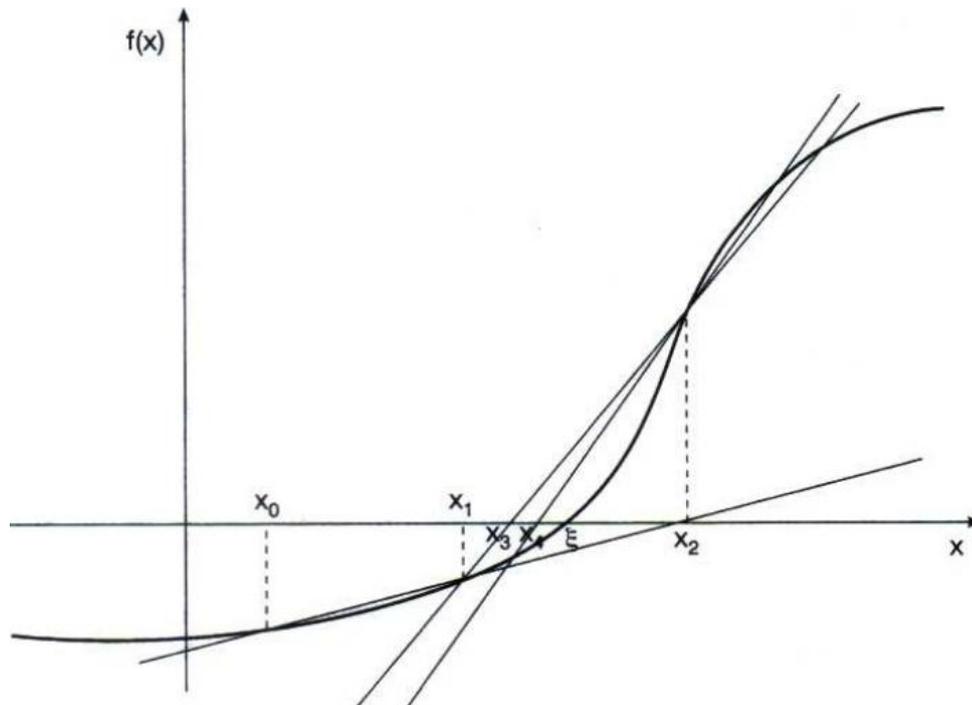
Esta seria a escolha deste trabalho, não fosse um pré-requisito do método: a necessidade de se calcular a derivada da função principal (que se deseja obter raízes).

Mesmo assim, há um artifício que pode substituir a derivada da função, artifício este que determina o Método da Secante: substitui-se a derivada  $f'(x_k)$  pelo quociente das diferenças apresentado por (3.43). Então, tem-se a expressão que apresenta o método das secantes em (3.44).

$$f'(x_n) = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}} \quad (3.43)$$

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \cdot f(x_n) \quad (3.44)$$

Figura 31 – Interpretação Geométrica do Método da Secante



Fonte: Ruggiero e Lopes (2006).

Tal alteração reduz a ordem de convergência do método, que segundo Ruggiero e Lopes (2006) é quadrática para o Método de Newton, e equivale a  $p=1,618\dots$  para o Método da Secante: também sendo satisfatório por ser maior que 1, que é a convergência linear.

A comparação escolhida para critério de parada é a do erro absoluto não é aconselhável para casos de números de grandes dimensões e erros muito pequenos, porém, não se trata do caso abordado neste trabalho, onde a solução está entre -10 e 3,5 (aproximadamente), explicando o motivo deste critério ser o escolhido. A equação que o define é apresentada em (3.45).

$$x_{k+1} - x_k = \epsilon \quad (3.45)$$

Em que  $\epsilon$  representa o erro absoluto aceitável para o processo iterativo, cujo valor padrão estipulado no programa foi de 0,00001, para todas as iterações, com exceção da iteração de verificação de estado-limites de fissuração, em que foi estipulado que a soma dos erros de todas as variáveis por padrão não ultrapassa 0,00001, que é mostrado mais explicitamente ao final da seção 3.8.3.

Além disso, menciona-se o método de Newton para a resolução de sistemas não lineares na equação (3.46), que se assemelha à equação (3.42), porém todas as variáveis são expressas em formas matriciais. Para aplicar o método, é necessária uma matriz chamada Jacobiana  $J_F$ , que é a matriz contendo as derivadas parciais de  $F(x)$ .

As matrizes contendo as variáveis são apresentadas por (3.47), o Jacobiano é apresentada por (3.48) e a matriz contendo as funções é apresentada (3.49).

$$x_{n+1} = x_n - \frac{F(x_n)}{J_F(x_n)} \quad (3.46)$$

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad (3.47)$$

$$J_F = \begin{pmatrix} \frac{\partial f_1(x)}{\partial x_1} & \frac{\partial f_1(x)}{\partial x_2} & \dots & \frac{\partial f_1(x)}{\partial x_n} \\ \frac{\partial f_2(x)}{\partial x_1} & \frac{\partial f_2(x)}{\partial x_2} & \dots & \frac{\partial f_2(x)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n(x)}{\partial x_1} & \frac{\partial f_n(x)}{\partial x_2} & \dots & \frac{\partial f_n(x)}{\partial x_n} \end{pmatrix} \quad (3.48)$$

$$F(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \\ f_3(x) \end{pmatrix} \quad (3.49)$$

Usando a denotação para  $x_{n+1} - x_n$  apresentada em (3.50), confirma-se que  $s_n$  é a solução do sistema linear apresentado por (3.51):

$$s_n = x_{n+1} - x_n \quad (3.50)$$

$$s_n = - \frac{F(x_n)}{J_F(x_n)} \quad (3.51)$$

Como descreve Ruggiero e Lopes (2006) e confirma-se ao analisar a equação (3.46), este processo necessita a resolução do sistema linear apresentado em (3.51). Além disso, é necessário avaliar a matriz Jacobiana em cada  $x_n$ , isto é, calcular os valores das derivadas parciais a cada iteração, tornando o método de Newton demasiadamente dispendioso para resolução de sistemas não lineares.

Assim, segunda mesma autora, cita-se a importância de outros métodos de resolução de sistemas não lineares: os métodos quase-Newton, que buscam alternativas que contornem a principal dificuldade do método de Newton, que consiste no cálculo da matriz Jacobiana a cada iteração.

Primeiramente, cita-se o método de Newton modificado: que consiste em calcular o Jacobiano apenas a primeira vez, na primeira iteração, e aplicar no cálculo de todas as etapas a mesma matriz Jacobiana.

Outro recurso citado pela mesma referência é o método de Newton discreto, que substitui as derivadas por uma aproximação numérica, substituindo os termos do jacobiano. Com base na definição de derivada, mostrada na equação (3.52), substitui-se os termos do jacobiano pelo apresentado em (3.53):

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}, \quad h \approx 0 \quad (3.52)$$

$$\frac{\partial f_i(x)}{\partial x_j} \approx \frac{f_i(x + he_j) - f_i(x)}{h}, \quad h \approx 0 \quad (3.53)$$

Sendo:

$e_j = (0, 0, \dots, 1, 0, \dots, 0)^T$ , onde o elemento 1 ocupa a posição  $j$ ;

A existência da matriz  $e_j$  diz respeito à variável independente da derivada que receberá o acréscimo, lembrando que na notação da equação (3.53),  $x$  simboliza uma matriz e não uma variável apenas.

Apresentados estes métodos para resolução de sistemas não lineares de equações, apresenta-se os conceitos e suas respectivas maneiras de aplicação no programa e sugestões de uso na seção 3.8.3.

### 3.8.1 Estado-limite último

A rotina de estado-limite último, sintetizando o raciocínio das seções 2.5 e 3.6, procura a conformação de deformações que apresente o valor de força axial desejado. Tais valores existem pra cada ângulo  $\alpha$  de inclinação da linha neutra. E as conformações de deformações diferentes são aqui apresentadas como  $\varepsilon_1$  e  $\varepsilon_2$ .

Escrevendo matematicamente o apresentado, tem-se (3.54).

$$N = \Phi_N(\alpha, \varepsilon_1, \varepsilon_2) \quad (3.54)$$

Primeiramente, leva-se em conta que  $N$  é um valor conhecido, e  $M_{xx}$  e  $M_{yy}$  são desconhecidos, e devem ser calculados, por isso suas expressões não são apresentadas. As conformações de deformações são de ruptura, portanto seguem o diagrama de domínio da norma de estruturas de concreto ABNT NBR 6118:2014, mostrado na seção 2.3, e por consequência estará incluso numa das regiões mostradas na mesma seção: as cargas axiais limítrofes dessas regiões podem ser calculadas, pois:

Região I: Entre reta A e reta que divide os domínios 2 e 3;

Região II: Reta que divide os domínios 2 e 3, e reta que divide os domínios 4 e 5;

Região III: Reta que divide os domínios 4 e 5, e reta B.

Seguindo o apresentado, adota-se um raciocínio que permite analisar o problema como uma equação não-linear com uma variável apenas, pois  $\alpha$  será constante passo-a-passo, isto é, não é um valor a ser encontrado (isto é, uma incógnita), mas uma variável que assumirá valores, assim como apresenta (3.55).

$$N_\alpha = \Phi_{N,\alpha}(\varepsilon_1, \varepsilon_2) \quad (3.55)$$

Já a conformação de deformações é uma dupla de incógnitas, podendo ser substituído por uma incógnita apenas, assumindo que:

Na região II  $\varepsilon_1$  é constante e  $\varepsilon_2$  varia. Se  $\varepsilon_1$  é constante, este não é mais uma variável de  $\Phi_N$ , pode-se assumir que  $\Phi_N$  é um função apenas de  $\varepsilon_2$ .

Na região III  $\varepsilon_2$  é constante e  $\varepsilon_1$  varia. Analogamente ao caso da região II, pode-se assumir que  $\Phi_{N,\alpha}$  é um função apenas de  $\varepsilon_1$ .

Na região I  $\varepsilon_2$  e  $\varepsilon_1$  variam. Porém, um pode ser função do outro, assim como mostra a Figura 4: as deformações devem ser tais que passem pelo ponto C. Porém, se o concreto usado for C90,  $\varepsilon_1$  é constante, pois  $\varepsilon_{cu} = \varepsilon_{c2}$ , então será considerado, para todos as resistências, uma função de  $\varepsilon_2$ , e  $\varepsilon_1$  será função deste seguindo a equação (3.57), obtida simplificando (3.27) considerando a posição do ponto C, isto é, o mostrado em (3.56). Portanto assume-se que  $\Phi_{N,\alpha}$  é um função apenas de  $\varepsilon_2$ .

$$d_{\varepsilon c2} = \frac{\varepsilon_{cu} - \varepsilon_{c2}}{\varepsilon_{cu}} \cdot h \quad (3.56)$$

$$\varepsilon_1 = \frac{\varepsilon_{c2} \cdot \varepsilon_{cu} - (\varepsilon_{cu} - \varepsilon_{c2}) \cdot \varepsilon_2}{\varepsilon_{c2}} \quad (3.57)$$

O método numérico procura uma raiz da função, isto é, um ponto tal que o resultado da função seja zero. Porém, a carga axial que deverá ser atingida pela iteração pode não ser zero. Para inclui-la no problema, basta subtrair seu valor no final da função  $\Phi_{N,\alpha}$ , que é dada, para uma certa conformação de deformações, a soma da força axial no concreto, força axial no aço, e no final subtrai a carga procurada.

Matematicamente, o problema segue o indicado em (3.58), e a função a ser desenvolvida no método das secantes é a (3.59).

$$N_c + N_s = N_{usuário} \quad (3.58)$$

$$N_c + N_s - N_{usuário} = 0$$

$$\Phi(\varepsilon_n) = N_c + N_s - N_{usuário} \quad (3.59)$$

Como expressão final pra obtenção da conformação de deformação, apresenta-se (3.60).

$$\varepsilon_{n+1} = \varepsilon_n - \frac{\varepsilon_n - \varepsilon_{n-1}}{\varnothing(\varepsilon_n) - \varnothing(\varepsilon_{n-1})} \cdot \varnothing(\varepsilon_n) \quad (3.60)$$

Além disso, recomenda-se que as rotinas sempre apresentem a altura da linha neutra obtida no ELU, para que o usuário sempre possa comparar com o limite imposto pela seção 14.6.4.3 da norma ABNT NBR 6118:2014. No presente trabalho, adotou-se calcular  $x/h$ , embora os valores de  $\varepsilon_1$  e  $\varepsilon_2$  já permitiram fazê-lo.

### 3.8.2 Gráfico Momento-Normal-Curvatura

Ao aplicar a situação mostrada em 3.7.1 no método das secantes, primeiramente recorda-se da situação a ser resolvida: atribui-se um valor de curvatura, e a partir deste valor procura-se a conformação de deformações, o que significa que a diferença entre  $\varepsilon_1$  e  $\varepsilon_2$  é fixa, logo apenas um destes valores é variável. Faz-se uma sugestão considerada especialmente útil: a variável a ser procurada é o  $\varepsilon_2$ , e o  $\varepsilon_1$  deverá responder a uma função deste.

Tal consideração vale a pena ao se analisar o que ocorre com a força axial nos limites de compressão: observando os pontos de ruptura B e C, mostrados na Figura 4, o ponto com maior carga axial N é o ponto C, em que  $\varepsilon_1 = \varepsilon_2 = \varepsilon_{c2}$ . E o ponto B corresponde à situação em que  $\varepsilon_1 = \varepsilon_{cu}$  e  $\varepsilon_2 = 0$ . Logo, a derivada ( $dN/d\varepsilon_1$ ) é positiva se o ponto de ruptura for B, mas negativa se o ponto de ruptura for C. Isto já não ocorre com  $\varepsilon_2$ : a derivada ( $dN/d\varepsilon_2$ ) é sempre positiva (ou nula, em seu menor valor).

Portanto, usando  $\varepsilon_2$  como referência, procura-se o valor de  $\varepsilon_2$  que resultará no N procurado. Recapitulando o método das secantes, mostrado na equação (3.44) e com explicação geométrica mostrada na Figura 31, apresenta-se a função que representa o processo numérico iterativo por (3.61):

$$\varepsilon_{2,n+1} = \varepsilon_{2,n} - \frac{\varepsilon_{2,n} - \varepsilon_{2,n-1}}{N(\varepsilon_{2,n}) - N(\varepsilon_{2,n-1})} \cdot N(\varepsilon_{2,n}) \quad (3.61)$$

O método das secantes é usado para obtenção das raízes da função: logo, caso se queira chegar à um valor  $N_u$  na função  $N(\varepsilon_2)$ , deve-se forçar a função a assumir esse valor, subtraindo  $N_u$  de  $N(\varepsilon_2)$ . Assim, o zero da função aplicada no método das secantes ocorrerá quando a força axial correspondente à conformação de deformações for  $N_u$ .

Outra alteração que foi necessária para se manter o funcionamento da função: ao se analisar (3.61), observa-se que  $N(\varepsilon_{2,n})$  e  $N(\varepsilon_{2,n-1})$  não podem ter o mesmo valor ou a formula apresenta um zero no denominador. Entretanto, há uma região que isso ocorre: quando a

deformação na seção inteira for menor que a tração de escoamento do aço, a força é constante entre  $\varepsilon_{yd}$  e  $\varepsilon_u$ , (na ausência da armadura de protensão), por isso é necessário incluir algumas linhas que não permitam que o método assumira tal situação.

Como citado, o processo iterativo termina quando a equação (3.45) é atendida, isto é, quando a diferença entre duas iterações consecutivas é menor que o erro absoluto admitido, cujo valor padrão é 0,00001, e sugere-se que este valor possa ser alterado pelo usuário, o que é seguido no programa.

### 3.8.3 Verificação de estados-limite de fissuração e descompressão

Para aplicar um método numérico ao problema representado pela expressão (3.41) tem-se uma dificuldade com as variáveis, que apresentam alterações no comportamento sob certas circunstâncias: por exemplo, uma alteração no ângulo  $\alpha$  de 180° inverte totalmente a influencia das variáveis  $\varepsilon_1$  e  $\varepsilon_2$  nos momentos. Esse comportamento altera totalmente o comportamento das funções e suas relações com as variáveis, o que não é desejado para aplicação do método numérico aqui utilizado.

Portanto, deseja-se uma perspectiva diferente de variáveis, de modo a tornar as funções mais estáveis: troca-se a perspectiva apresentada por uma combinação de variáveis que considere a deformação no ponto médio da seção e as curvaturas em cada direção: Já a curvatura pode contornar este problema sob certas circunstâncias:

Considerando a curvatura em módulo, esta pode ocorrer em qualquer direção, portanto para se obter um comportamento mais previsível das respostas que uma alteração na curvatura possa acarretar nos momentos, esta é separada em duas componentes: curvatura em  $x$  e curvatura em  $y$ .

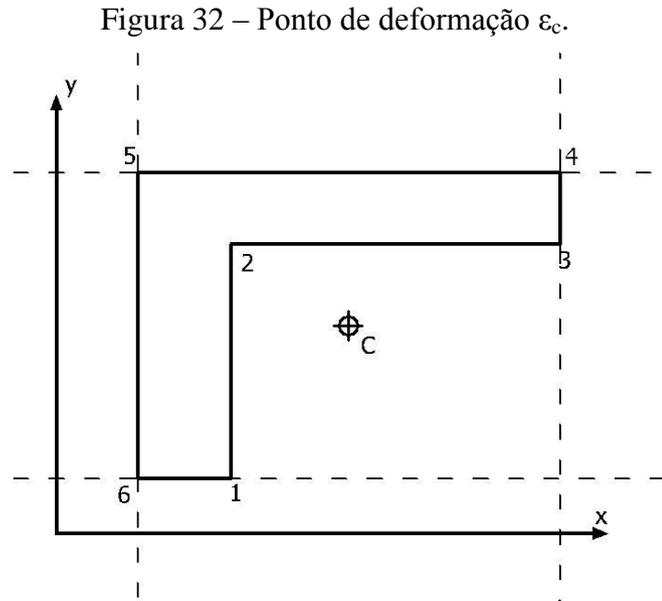
Na literatura, Leonhardt e Mönnig (1979) representa a curvatura pela letra grega  $\kappa$ , que será usada também neste trabalho, além disso, representa-se a curvatura  $x$  por  $\kappa_x$ , apresentada pela equação (2.14), e a curvatura  $y$  por  $\kappa_y$ , apresentada pela equação (2.15). Apresenta-se mais sobre esses conceitos na seção 2.6.1 e em suas fontes.

Dito isso, apresentam-se as duas variáveis independentes da iteração proposta:  $\Delta\varepsilon_x$  e  $\Delta\varepsilon_y$ , que são definidas respectivamente pelas equações (3.62) e (3.63) a seguir.

$$\kappa_x = \frac{1}{r_x} = \frac{\varepsilon_{x,o} - \varepsilon_{x,u}}{h_x} = \frac{\Delta\varepsilon_x}{h_x} \quad (3.62)$$

$$\kappa_y = \frac{1}{r_y} = \frac{\varepsilon_{y,o} - \varepsilon_{y,u}}{h_y} = \frac{\Delta\varepsilon_y}{h_y} \quad (3.63)$$

A outra variável independente é denominada  $\varepsilon_{cr}$ : esta determina a deformação no ponto CR: o ponto no centro do retângulo circunscrito à seção. Isto é, com coordenada  $x$  sendo a média entre o menor e o maior valor em  $x$ , e analogamente em  $y$ , isto é, é um ponto localizado num centro referente não à seção como uma área, mas seus pontos extremos em cada coordenada. A Figura 32 apresenta este ponto sob legenda C.



Assim, são definidas as 3 variáveis independentes que são aplicadas no método quase-Newton:  $\varepsilon_{cr}$ ,  $\Delta\varepsilon_x$  e  $\Delta\varepsilon_y$ . As variáveis dependentes são  $N$ ,  $M_{xx}$  e  $M_{yy}$ . Conhecendo essas variáveis é possível apresentar o Jacobiano, que é expresso por (3.64):

$$J_F = \begin{pmatrix} \frac{\partial N}{\partial \varepsilon_{cr}} & \frac{\partial N}{\partial \Delta\varepsilon_x} & \frac{\partial N}{\partial \Delta\varepsilon_y} \\ \frac{\partial M_{xx}}{\partial \varepsilon_{cr}} & \frac{\partial M_{xx}}{\partial \Delta\varepsilon_x} & \frac{\partial M_{xx}}{\partial \Delta\varepsilon_y} \\ \frac{\partial M_{yy}}{\partial \varepsilon_{cr}} & \frac{\partial M_{yy}}{\partial \Delta\varepsilon_x} & \frac{\partial M_{yy}}{\partial \Delta\varepsilon_y} \end{pmatrix} \quad (3.64)$$

Porém, assim como nas outras situações, é indesejado o cálculo da função derivada devido ao excesso de termos e condições, por exemplo, a existência ou não de armaduras ativas ou passivas, entre outros fatores.

Além disso, a aproximação permite um controle maior das variáveis (melhor descrito a seguir), o que é desejável neste trabalho, pois as rotinas podem apresentar comportamentos indesejados quando usadas incorretamente, podendo levar o programa a parar de funcionar, por exemplo. Pode-se citar como uso incorreto seria aplicar na rotina de força no concreto uma deformação maior que  $\varepsilon_{cu}$  (cujo padrão alterável é 3,5‰).

Os métodos numéricos citados não consideram inicialmente estes problemas em suas rotinas, pois as funções matemáticas muitas vezes tem um domínio infinito. Portanto o programador deve estar atento ao comportamento do processo iterativo e a maneira de aplicá-lo. Uma maneira de fazê-lo é como se apresenta a seguir:

Anteriormente foi citado o controle maior das variáveis. Este faz-se necessário no presente trabalho pois os valores das deformações não devem ultrapassar certos limites, podendo causar instabilidade das rotinas do programa. Logo, faz-se uma alteração matemática para que os saltos entre as iterações sejam pequenos: Um alto valor na derivada faz com que os saltos ( $s_n$ ) entre os valores  $x_n$  e  $x_{n+1}$  sejam pequenos. Isso é realizado exagerando o valor do Jacobiano (que substitui a derivada nos sistemas de equações), pois seguindo a lógica da explicação geométrica do Método de Newton (ver Figura 30 e equações (3.43) e (3.46) ) uma reta tangente muito inclinada resulta num salto pequeno.

Naturalmente, se os esforços inseridos pelo usuário excederem o limite dos materiais, é desejável que o processo de cálculo reconheça essa situação que extrapola o ELU, e a rotina interrompa o cálculo e o avise o usuário sobre o ocorrido.

Também é importante se destacar que a bibliografia apresenta certa divergência ao abordar esse cálculo: a maior parte dos trabalhos aborda o assunto dando segmento ao apresentado Santos (1994), mas na adoção das 3 variáveis que definem a conformação de deformações do plano admitem diferentes escolhas: Medeiros (2004) apresenta as variáveis  $\varepsilon_0$ ,  $\kappa_x$  e  $\kappa_y$ , Ceccon (2008) utiliza as variáveis  $\alpha$ ,  $(1/r_\alpha)$  e  $\varepsilon_0$ .

Ao se citar as rotinas para obter a solução de sistemas de equações não lineares, apresentou-se o método de Newton modificado e o método de Newton discreto (ver seção 3.8), e sugere-se que ambos sejam usados. Primeiramente, é desejável que a matriz Jacobiana seja constante para evitar certas inconsistências da rotina, que pode assumir valores inválidos em certas situações (por exemplo, quando  $\varepsilon < -\varepsilon_{yd}$ , nos casos em que não há protensão, qualquer alteração na deformação não acarreta alteração nas forças normais análogas), além disso, reduz-se a necessidade de se calcular a matriz jacobiana a cada iteração, o que é oneroso computacionalmente.

A segunda sugestão é que os termos da matriz Jacobiana sejam substituídos pelo cálculo discretizado, isto é, os termos apresentados na por (3.64) serão substituídos segundo o padrão apresentado pela equação (3.53). A condição inicial  $x_1$  é dada por (3.65):

$$x_1 = \begin{pmatrix} \varepsilon_c \\ \Delta\varepsilon_x \\ \Delta\varepsilon_y \end{pmatrix} = \begin{pmatrix} 0,05 \\ 0 \\ 0 \end{pmatrix}, \quad h = 0,1 \quad (3.65)$$

A matriz Jacobiana também parte de tal ponto. Tal valor é escolhido para que a seção apresente compressão e no ponto em que a força do concreto apresenta maior derivada (que é próximo de zero, como indica a Figura 9). Tal escolha de valores tem como objetivo proporcionar um valor alto para as derivadas, assim como citado. O resultado encontra-se na Tabela 2.

Retomando a matriz jacobiana a ser calculada, equação (3.64), e a equação do método de Newton discretizado, equação (3.53) com  $h=0,1$ , apresenta-se os termos a serem usados na matriz Jacobiana na Tabela 2 a seguir:

Tabela 2. Termos da matriz Jacobiana em função de  $\varepsilon_{cr}$ ,  $\Delta\varepsilon_x$  e  $\Delta\varepsilon_y$ .

$\frac{\partial N}{\partial \varepsilon_c} = \frac{N(0,15, 0, 0) - N(0,05, 0, 0)}{0,1}$
$\frac{\partial N}{\partial \varepsilon_c} = \frac{N(0,15, 0, 0) - N(0,05, 0, 0)}{0,1}$
$\frac{\partial M_{xx}}{\partial \varepsilon_c} = \frac{M_{xx}(0,15, 0, 0) - M_{xx}(0,05, 0, 0)}{0,1}$
$\frac{\partial M_{yy}}{\partial \varepsilon_c} = \frac{M_{yy}(0,15, 0, 0) - M_{yy}(0,05, 0, 0)}{0,1}$
$\frac{\partial N}{\partial \Delta\varepsilon_x} = \frac{N(0,05, 0,1, 0) - N(0,05, 0, 0, 0)}{0,1}$
$\frac{\partial M_{xx}}{\partial \Delta\varepsilon_x} = \frac{M_{xx}(0,05, 0,1, 0) - M_{xx}(0,05, 0, 0, 0)}{0,1}$
$\frac{\partial M_{yy}}{\partial \Delta\varepsilon_x} = \frac{M_{yy}(0,05, 0,1, 0) - M_{yy}(0,05, 0, 0, 0)}{0,1}$
$\frac{\partial N}{\partial \Delta\varepsilon_y} = \frac{N(0,15, 0, 0, 0) - N(0,05, 0, 0, 0)}{0,1}$
$\frac{\partial M_{xx}}{\partial \Delta\varepsilon_y} = \frac{M_{xx}(0,15, 0, 0, 0) - M_{xx}(0,05, 0, 0, 0)}{0,1}$
$\frac{\partial M_{yy}}{\partial \Delta\varepsilon_y} = \frac{M_{yy}(0,15, 0, 0, 0) - M_{yy}(0,05, 0, 0, 0)}{0,1}$

A rotina elaborada para o cálculo de  $N$ ,  $M_{xx}$  e  $M_{yy}$  atuantes na seção refere-se às variáveis  $\alpha$ ,  $\varepsilon_1$  e  $\varepsilon_2$ . Porém, as funções anteriormente são apresentadas em função de  $\varepsilon_{cr}$ ,  $\Delta\varepsilon_x$  e

$\Delta\varepsilon_y$ . No intuito de se calcular a matriz de funções ( $N$ ,  $M_{xx}$ ,  $M_{yy}$ ) em relação aos termos ( $\varepsilon_{cr}$ ,  $\Delta\varepsilon_x$  e  $\Delta\varepsilon_y$ ) apresenta-se a seguinte dedução, considerando  $G$ ,  $H$  e  $J$  funções matriciais:

No intuito de se calcular (3.66), considerando que as funções para se calcular (3.67) já são conhecidas, apresenta-se a função composta adotada na equação (3.68).

$$\begin{pmatrix} N \\ M_{xx} \\ M_{yy} \end{pmatrix} = G \begin{pmatrix} \varepsilon_{cr} \\ \Delta\varepsilon_x \\ \Delta\varepsilon_y \end{pmatrix} \quad (3.66)$$

$$\begin{pmatrix} \alpha \\ \varepsilon_1 \\ \varepsilon_2 \end{pmatrix} = H \begin{pmatrix} \varepsilon_{cr} \\ \Delta\varepsilon_x \\ \Delta\varepsilon_y \end{pmatrix} \quad (3.67)$$

$$\begin{pmatrix} N \\ M_{xx} \\ M_{yy} \end{pmatrix} = J \begin{pmatrix} \alpha \\ \varepsilon_1 \\ \varepsilon_2 \end{pmatrix} = J \circ H \begin{pmatrix} \varepsilon_{cr} \\ \Delta\varepsilon_x \\ \Delta\varepsilon_y \end{pmatrix} \leftrightarrow G = J \circ H \quad (3.68)$$

E finalizando as deduções, apresenta-se  $H$ , isto é, as funções que determinam  $\alpha$ ,  $\varepsilon_1$  e  $\varepsilon_2$  a partir dos valores de ( $\varepsilon_{cr}$ ,  $\Delta\varepsilon_x$  e  $\Delta\varepsilon_y$ ). Como indicado nesta mesma seção,  $\varepsilon_c$  representa a deformação no centro do retângulo circunscrito,  $\Delta\varepsilon_x$  representa a diferença na deformação entre o ponto de maior coordenada  $x$  e o ponto de maior coordenada  $y$  e  $\Delta\varepsilon_y$  é semelhante à  $\Delta\varepsilon_x$ , porém referente à coordenada  $y$ . Tais variáveis são mais bem definidas na Figura 32 e no texto que a antecede.

Analisando as variáveis, verifica-se que é possível aplica-las nas equações (2.14) e (2.15), que calculam as curvaturas em relação à  $x$  e  $y$ , respectivamente, ( $1/r_x$ ) e ( $1/r_y$ ), para isso, é necessário calcular o comprimento em cada direção  $h_x$  e  $h_y$ . Então, demonstra-se (3.69) tomando (2.20) e (2.21), elevando-as ao quadrado, somando-as e aplicando a identidade trigonométrica ( $\text{sen}^2 x + \text{cos}^2 x = 1$ ).

$$\frac{1}{r_x^2} + \frac{1}{r_y^2} = \frac{1}{r_\alpha^2} \quad (3.69)$$

Além disso, o ângulo da ocorrência da curvatura  $\alpha$  (ver Figura 14) pode ser avaliado com base em (3.70), obtida dividindo-se (2.20) por (2.21) e isolando-se  $\alpha$ . Naturalmente, também deve-se avaliar seu quadrante, pois a função  $\text{arctg}$  retorna valores:  $-90 < \text{arctg} \leq 90$ .

$$\alpha = \text{arctg} \left( \frac{\frac{1}{r_x}}{\frac{1}{r_y}} \right) \quad (3.70)$$

Sabendo-se o ângulo  $\alpha$ , o objetivo atual é calcular  $\varepsilon_1$  e  $\varepsilon_2$ . Para obtê-los, parte-se dos valores de  $\varepsilon_{cr}$ ,  $\Delta\varepsilon_x$  e  $\Delta\varepsilon_y$ , inicialmente. Também conhece-se os valores de  $x_{max}$ ,  $x_{min}$ ,  $y_{max}$ ,  $y_{min}$  e as coordenadas  $xy$  do ponto em que ocorre  $\varepsilon_c$ , isto é, o centro do retângulo circunscrito.

Conhecendo-se  $\partial\varepsilon/d_x$ ,  $\partial\varepsilon/d_y$  e o valor  $\varepsilon$  em qualquer ponto, pode-se calcular o valor  $\varepsilon$  de qualquer outro ponto. O ponto cuja deformação se conhece é o  $\varepsilon_{cr}$ , e suas coordenadas devem também ser memorizadas. Uma alternativa é usar a equação (3.71).

$$\varepsilon = \varepsilon_{cr} + \frac{\partial\varepsilon}{\partial x} \cdot (x - x_{cr}) + \frac{\partial\varepsilon}{\partial y} \cdot (y - y_{cr}) \quad (3.71)$$

Outra alternativa é primeiramente girar a seção pelo ângulo  $\alpha$  e também girar o ponto CR em relação ao mesmo ponto de referência e sob o mesmo ângulo. Então, calcula-se a distância entre o ponto CR e as coordenadas máximas em  $v$  (isto é,  $y$  da seção girada por  $\alpha$ ). O valor de  $\varepsilon_1$  e  $\varepsilon_2$  é calculado conhecendo-se a variação entre eles  $\Delta\varepsilon = \varepsilon_1 - \varepsilon_2$  e considerando as diferenças de deformação proporcionais à distância, que analisando, é igual a curvatura, sendo a curvatura na direção  $u$  (perpendicular à  $v$ ) igual à zero.

$$\frac{\varepsilon_1 - \varepsilon_{cr}}{v_{max} - v_{cr}} = \frac{\varepsilon_2 - \varepsilon_{cr}}{v_{min} - v_{cr}} = \frac{\partial\varepsilon_v}{\partial v} = \frac{1}{r} \quad (3.72)$$

Ao invés de se aplicar a equação, deduz-se sua forma nas coordenadas  $uv$ , isto é, na seção girada por um ângulo  $\alpha$ . Sabendo-se que  $\partial\varepsilon_x/\partial x = (1/r_x)$ ;  $\partial\varepsilon_y/\partial y = (1/r_y)$ ;  $\partial\varepsilon/d_v = (1/r)$ . Aplica-se essas relações na equação (3.71), e realiza-se as substituições realizadas em (2.20) e (2.21), e então obtém-se (3.73) Nesta, considera-se a equação de rotação de pontos apresentada pela equação (3.10), e obtém-se (3.74).

$$\varepsilon = \varepsilon_{cr} + \left(\frac{1}{r_\alpha}\right) \cdot \left(x \cdot \cos \alpha + y \cdot \sin \alpha - \frac{1}{r_\alpha} \cdot \cos \alpha + \frac{1}{r_\alpha} \cdot \sin \alpha\right) \quad (3.73)$$

$$\varepsilon = \varepsilon_{cr} + \left(\frac{1}{r_\alpha}\right) \cdot (v - v_{cr}) \quad (3.74)$$

Ressalta-se que  $\partial\varepsilon/\partial u = 0$ , sendo  $u$  perpendicular à  $v$ .

Finalmente, a partir dos valores valores de  $v_{max}$  e  $v_{min}$  calculam-se, respectivamente,  $\varepsilon_1$  e  $\varepsilon_2$ .

Aplicam-se os valores de  $\alpha$ ,  $\varepsilon_1$  e  $\varepsilon_2$  nas rotinas já apresentadas neste capítulo, e obtém-se os valores de  $N$ ,  $M_{xx}$  e  $M_{yy}$  que permitem continuar as operações.

#### 4 VALIDAÇÃO DOS RESULTADOS

Para mostrar os resultados do programa, e avaliar sua confiabilidade, foram calculados alguns exemplos e comparados à bibliografia utilizada. O nome do programa que reúne as rotinas apresentadas nesta dissertação é OblqCALCO. Para obter informações sobre a maneira de se utilizá-lo, sugere-se consultar os apêndices.

No exemplo 1, considerou-se o dimensionamento realizado por Carvalho e Figueiredo Filho (2007): a seção e sua armadura, que foram calculadas para uma situação de flexão simples. A armadura aplicada no programa é exatamente a obtida pelo dimensionamento (não adaptada às bitolas comerciais), e então se compara os valores obtidos por este trabalho (com o programa OblqCALCO). Então, o mesmo resultado foi comparado ao dimensionamento do programa CALCO – Concreto Armado versão 4 (2009).

No exemplo 2, realizou-se o dimensionamento com auxílio dos ábacos de Pinheiro, Baraldi e Porem (2009). A armadura calculada é considerada na seção, cuja envoltória de momentos resistentes é gerada pelo programa. No mesmo diagrama é apresentado o momento usado para cálculo.

No exemplo 3, também foi realizado dimensionamento com auxílio dos ábacos de Pinheiro, Baraldi e Porem (2009), usando uma conformação de armaduras diferentes na seção, portanto, usando outro ábaco. Assim como o exemplo 2, apresenta-se o diagrama de momentos resistentes e o ponto que define o momento de dimensionamento.

No exemplo 4 comparam-se os gráficos de momentos resistentes de uma seção “L” gerados a partir deste trabalho e comparados à bibliografia Santos (1994), são gerados gráficos para duas forças normais diferentes: um de compressão e outro de tração.

No exemplo 5 são apresentados os resultados do programa deste trabalho e do programa Obliqua 1.0 (2001) para uma seção vazada, para avaliar se seus comportamentos são semelhantes.

O exemplo 6 avalia os valores da relação momento-curvatura apresentado por este trabalho, apresentado juntamente com os valores apresentados programa PCalc 1.1 (2013), confrontando valores apresentados com  $0,85 f_{cd}$  e  $1,1 f_{cd}$ .

O exemplo 7 compara a relação momento-curvatura com o trabalho de Ribeiro (2011), convertendo a variável curvatura  $1/r$  para curvatura adimensional  $\theta$ . Três relações momento-curvatura adimensional são apresentadas para uma seção retangular sob três forças normais diferentes.

O exemplo 8 apresenta uma seção de laje, cuja protensão não é considerada pelo programa, mas convertida em carregamento distribuído equivalente. Então, foi avaliada quanto à ELS-F e ELS-D.

O exemplo 9 analisa um exemplo proposto por Faleiros Junior (2010) de uma seção composta de elementos com duas resistências diferentes e armadura passiva e ativa, e verifica o ELS-W desta. O exemplo é resolvido pelas rotinas deste trabalho, considerando a protensão diretamente.

O exemplo 10 apresenta o dimensionamento de uma seção em concreto C90, e posterior avaliação do diagrama de momentos resistentes, para se avaliar o momento máximo na direção principal.

No exemplo 11 se avalia uma análise em ELU quanto à flexão com utilização de armadura ativa. A bibliografia fonte do exemplo analisado é Carvalho (2012).

O exemplo 12 analisa o estado-limite de serviço de abertura de fissuras, comparando com um exemplo da referência Carvalho e Figueiredo Filho (2007) – num caso de flexão simples.

#### 4.1 EXEMPLO 1 – FLEXÃO SIMPLES

Nesse exemplo, Carvalho e Figueiredo Filho (2007) calcula a armadura de uma seção em concreto armado, solicitada por um momento de 12,2 kNm. A seção tem  $b_w = 12$  cm e  $d = 29$ cm, concreto C20 e aço CA-50, logo trata-se de um caso de flexão simples, logo, a força axial  $N = 0$ . Portanto, o momento de cálculo é o produto do momento característico pelo coeficiente de segurança padrão para cargas de edifícios de 1,4 em estado-limite último 1,4, resultando num momento de cálculo 17,08 kNm.

Com o dimensionamento com auxílio do KMD, o autor chega a uma armadura teórica  $A_s = 1,46$  cm<sup>2</sup>. Não se trata de uma área de bitola comercial, mas mesmo assim, o programa pode realizar o cálculo:

Para realizar, introduz-se no programa uma seção de 12 cm x 32 cm, e apenas uma armadura, na altura de 3 cm, forçando a atribuição do autor de  $d = 29$ cm. Além disso, é preciso introduzir a resistência característica à compressão do concreto para 20 MPa.

O programa com os dados usados é apresentado na Figura 33. Os pontos que definem a seção de concreto formam uma seção retangular, com as coordenadas: ( -6 , 0 ); ( 6 , 0 ); (6,32); ( -6,32 ); a coordenada da armadura passiva é (0,3) e sua área é de 1,46cm<sup>2</sup>. A força axial introduzida é nula, pois trata-se de um caso de flexão simples. Pelo mesmo motivo, analisa-se o valor com ângulo  $\alpha = 0$ , isto é, com a linha-neutra na posição horizontal padrão.

Figura 33 – Programa elaborado, OblqCALCO, com os dados de análise.

The screenshot shows the OblqCALCO software interface. At the top, there is a menu bar with 'Arquivo' and 'Ajuda'. Below the menu bar, there is a 'Materiais' button. The main area contains several input fields and checkboxes. The 'Armadura Passiva' checkbox is checked, and 'Armadura Ativa' is unchecked. The 'Número de pontos' is set to 4, 'Número Barras' is set to 1, and 'Número Barras' is also set to 1. There are three tables: the first table shows coordinates (X, Y) for 4 points; the second table shows coordinates (x, y) and area (As) for 1 bar; the third table shows coordinates (x, y) for 2 bars. Below the tables, there are tabs for 'ELU', 'Momento-Curvatura', 'Verificar Solic. ELS', and 'Calcular Solicitacoes'. The 'Calcular Solicitacoes' tab is active, showing input fields for 'Nd (kN)', 'Mxx (kN.cm)', and 'Myy (kN.cm)', all set to 0. A 'Calcular' button is present. Below the input fields, there is a text box with the text 'Lembrar: Compressão positiva; Tração negativa.' and a text box for results. The results show 'Nmax (kN)' as 527,60571 and 'Nmin (kN)' as -63,478261.

Arquivo Ajuda

Materiais

Armadura Passiva  Armadura Ativa

Número de pontos: 4 Número Barras 1 Número Barras

	X (cm)	Y (cm)
1	-6	0
2	6	0
3	6	32
4	-6	32

	x (cm)	y (cm)	As (cm <sup>2</sup> )
1	0	3	1,46

	x (cm)	y (cm)
1	-7	-17
2	7	-17

ELU Momento-Curvatura Verificar Solic. ELS Calcular Solicitacoes

Nd (kN) 0 Mxx (kN.cm) 0 Myy (kN.cm) 0 Calcular

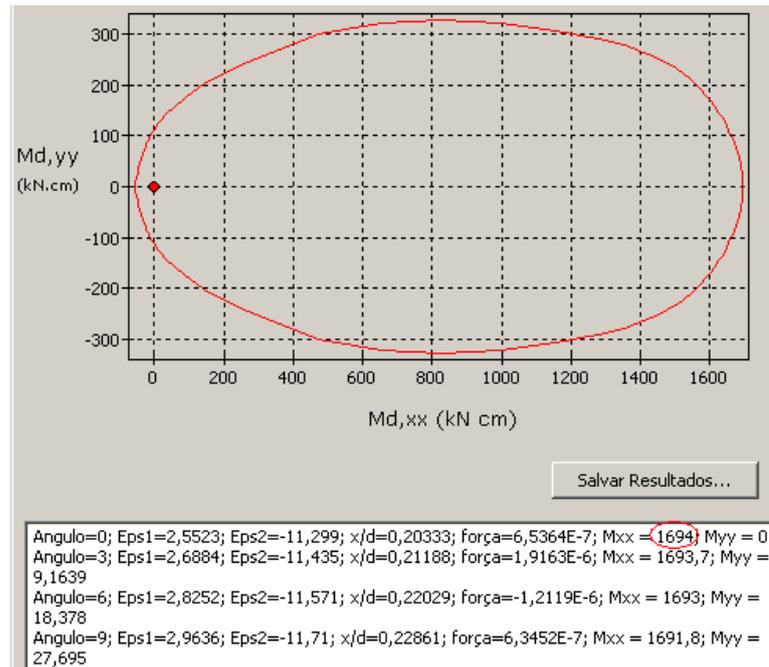
Lembrar: Compressão positiva; Tração negativa.

Nmax (kN) 527,60571

Nmin (kN) -63,478261

Os resultados do programa estão apresentados na Figura 34 e posteriormente citados e comparados.

Figura 34 - Resultados do exemplo 1, segundo o programa elaborado.



Assim:

Momento segundo a bibliografia: 17,08 kNm.

Momento limite segundo o programa: 16,94 kNm

Variação entre os momentos:  $16,94/17,08 = 0,008 = 0,8\%$

Outra comparação é com o programa CALCO – Concreto Armado versão 4. O programa calcula a armadura passiva de uma seção submetida à flexão simples. Inserindo os dados deste exemplo, com exceção da armadura passiva, o programa fornece o valor de armadura com a qual a seção deve ser armada. Os dados de entrada e os resultados do programa são mostrados na Figura 35.

Figura 35 – Armadura do Exemplo 1, segundo programa de cálculo de armadura.

Dados de Entrada - ELU				Resultados			
Mg <sub>1</sub>	γ <sub>f, g<sub>1</sub></sub>	Mg <sub>2</sub>	γ <sub>f, g<sub>2</sub></sub>	A <sub>s</sub>		M <sub>d</sub>	
12,2	1,4	0	0	1,47	cm <sup>2</sup>	17,08	kN.m
Mq	γ <sub>f, q</sub>			KX (x/d)	0,205	x =	0,059 m
0	0			KZ (z/d)	0,92	z =	0,267 m
		CA 50	γ <sub>s</sub> = 1,15	Sub-domínio 2b			
bw = 0,12 m	f <sub>ck</sub> = 20 MPa			ε <sub>c</sub>	2,57 ‰	ε <sub>s</sub>	10 ‰
h = 0,32 m	γ <sub>c</sub> = 1,4			Mdmax (kN.m)	39,29	DMIN (m)	0,19
dt = 0,29 m	x/d lim = 0,5						

Como a armadura calculada pelo programa ( $1,47\text{cm}^2$ ) foi bastante próximo da armadura utilizada para o mesmo momento fletor ( $1,46\text{cm}^2$ ), considera-se a análise realizada pelo programa válida.

#### 4.2 EXEMPLO 2 – FLEXÃO COMPOSTA OBLÍQUA

Neste item, apresenta-se o dimensionamento realizado segundo os ábacos de Pinheiro, Baraldi e Porem (2009), e depois, sua verificação no programa com as rotinas apresentadas.

Lembrando que a notação usada para os momentos no trabalho atual e nesta bibliografia específica apresentam notação semelhante e significado diferente, portanto recomenda-se que o leitor seja atento a imagem apresentada para a seção, na Figura 36, que facilita grandemente a compreensão do problema. A seção possui os seguintes materiais, dimensões e solicitações:

$$b_w = 30\text{cm}; h = 45\text{ cm}; f_{ck} = 25\text{ MPa.}$$

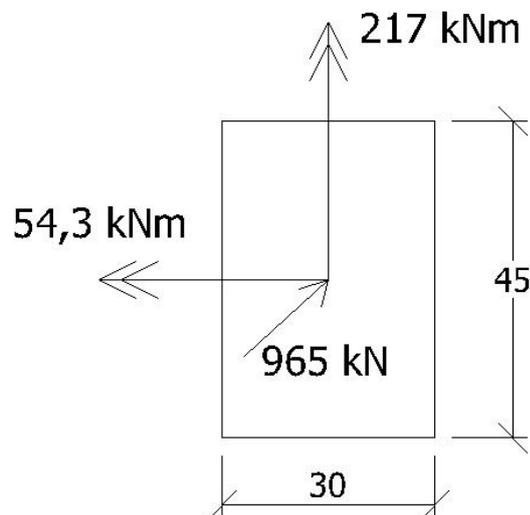
$$N_d (\text{força axial}) = 965\text{ kN}$$

$$M_{xx} (\text{momento em torno do eixo } x) = 54,3\text{kNm} (\text{ou excentricidade } e_y \text{ igual a } 5,627\text{ cm})$$

$$M_{yy} (\text{momento em torno do eixo } y) = 217\text{kNm} (\text{ou excentricidade } e_x \text{ igual a } 22,49\text{ cm})$$

Apresenta-se a seção e seus esforços solicitantes na Figura 36 a seguir:

Figura 36 – Seção a ser dimensionada no exemplo 2.



Para realizar o dimensionamento, deve-se consultar nos ábacos apresentados por Pinheiro, Baraldi e Porem (2009). Os ábacos são expressos em termos adimensionais. Trata-se dos esforços reduzidos  $v$ ,  $\mu_x$  e  $\mu_y$  e da taxa mecânica de armadura  $\omega$ . Nos ábacos, procura-se pela taxa mecânica de armadura correspondente aos esforços adimensionais  $v$ ,  $\mu_x$  e  $\mu_y$ .

Portanto, inicialmente, calcula-se o das três variáveis, cujas equações são mostradas por (4.1), (4.2) e (4.3). Depois, procura-se nos ábacos pelo valor  $\omega$  correspondente, e calcula-se a área de armadura passiva pela equação (4.4)

$$v = \frac{N_d}{A_c \cdot f_{cd}} \quad (4.1)$$

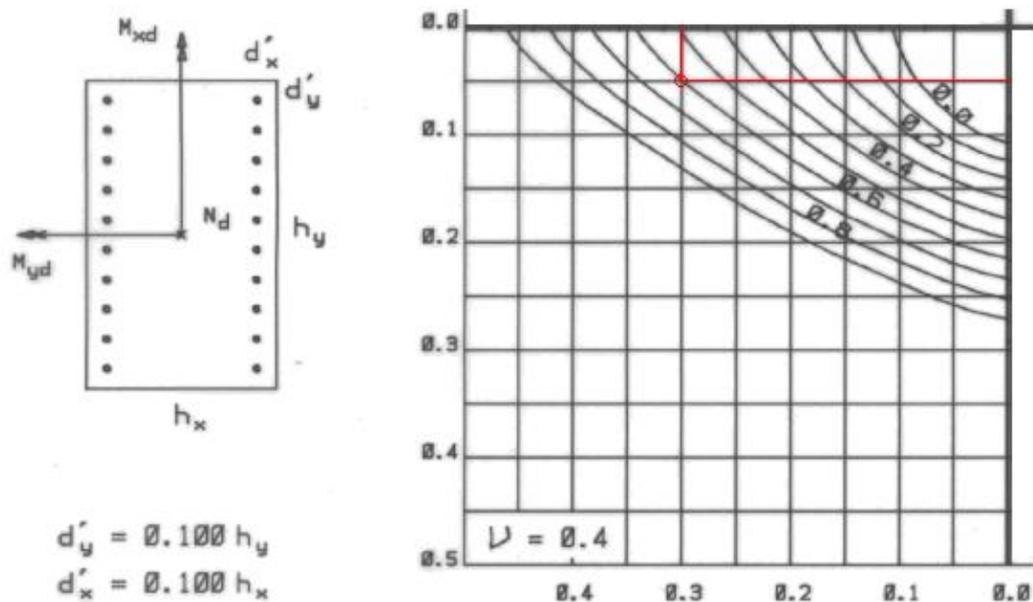
$$\mu_x = \frac{M_{yy}}{A_c \cdot h_x \cdot f_{cd}} \quad (4.2)$$

$$\mu_y = \frac{M_{xx}}{A_c \cdot h_y \cdot f_{cd}} \quad (4.3)$$

$$\omega = \frac{A_s \cdot f_{yd}}{A_c \cdot f_{cd}} \rightarrow A_s = \frac{\omega \cdot A_c \cdot f_{cd}}{f_{yd}} \quad (4.4)$$

Logo, primeiramente obtém-se os resultados dos 3 esforços adimensionais, cujos resultados são dados por:  $v = 0,4$ ;  $\mu_x = 0,3$  e  $\mu_y = 0,05$ . Então, procura-se o ábaco que condiz com a geométrica da seção. Trata-se do ábaco 39 da bibliografia Pinheiro, Baraldi e Porem (2009). Neste trabalho, apresentado parcialmente pela Figura 37

Figura 37 – Ábaco usado para resolução do exemplo 2.



Fonte: Pinheiro, Baraldi e Porem (2009).

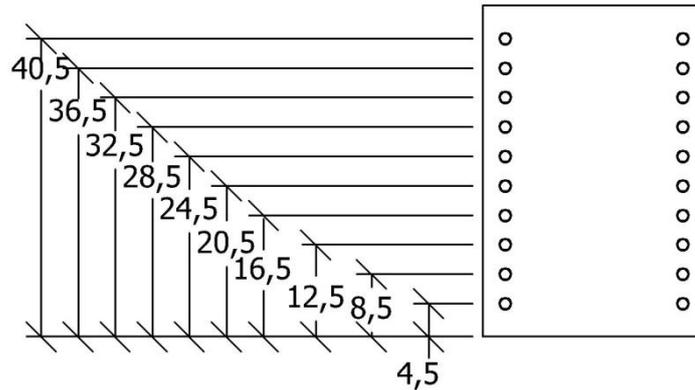
Ao se consultar o ábaco, chega-se à taxa mecânica de armadura  $\omega = 0,6$ .

$$A_s = \frac{\omega \cdot A_c \cdot f_{cd}}{f_{yd}} = \frac{0,6 \cdot 30 \cdot 45 \cdot 1,7857}{43,478} = 33,268 \text{ cm}^2$$

Adotou-se 20 barras cuja área é 1,6634, para a seção apresentar exatamente a área calculada.

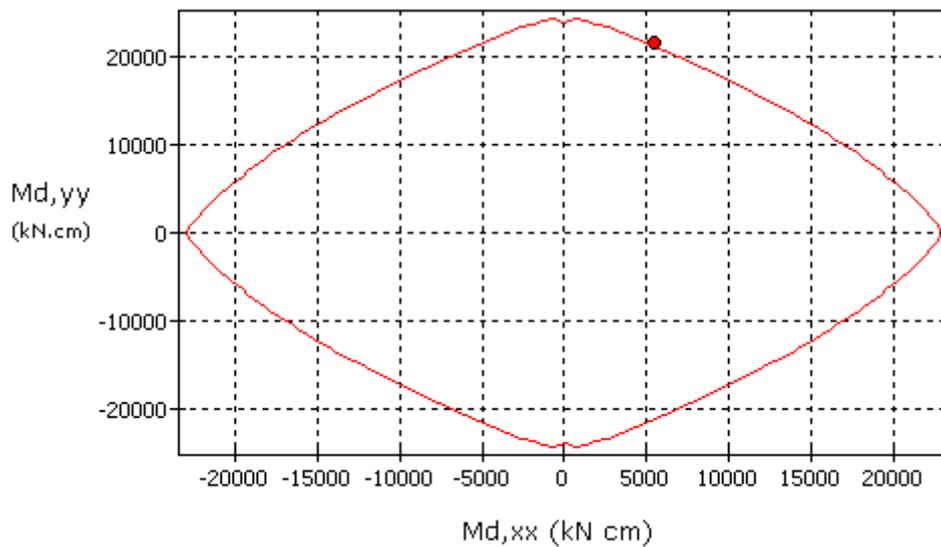
Para testar a seção calculada no programa, é necessário que se conheça a posição da armadura, portanto representa-se o resultado obtido, e as alturas relativas de cada armadura na Figura 38 e a distância do centro da barra até a lateral da peça é de 3 cm. Alerta-se o leitor que não foram considerados cobrimento e armadura transversal, pois estes não são parte do escopo deste exemplo.

Figura 38 – Altura da armadura passiva dimensionada da seção do exemplo 2.



Finalmente, introduzindo a seção no programa e caracterizando seus materiais, efetua-se o cálculo do programa e obtêm-se os resultados apresentados pela Figura 39.

Figura 39 – Resultados do exemplo 2 pelo OblqCALCO.



Assim, considera-se um resultado razoável, principalmente considerando que os resultados obtidos graficamente estão sujeitos à pequenos erros de leitura.

#### 4.3 EXEMPLO 3 – FLEXÃO COMPOSTA OBLÍQUA

Ainda dimensionando de acordo com os ábacos de Pinheiro, Baraldi e Porem (2009), apresenta-se outro dimensionamento de uma seção sob esforços de flexão oblíqua composta e posterior avaliação dos resultados deste programa. Então, apresenta-se o diagrama de momentos resistentes para a mesma seção segundo programa com as rotinas dispostas no presente texto novamente. Segue os dados usados para o dimensionamento:

$$b_w = 20\text{cm}; d = 30\text{cm}; f_{ck} = 25 \text{ MPa.}$$

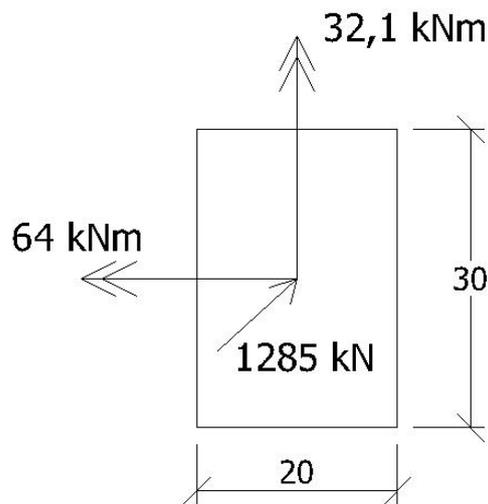
$$N_d \text{ (força axial)} = 1285 \text{ kN}$$

$$M_{xx} \text{ (momento em torno do eixo x)} = 64,0 \text{ kNm; ou } e_y = 0,05\text{m}$$

$$M_{yy} \text{ (momento em torno do eixo y)} = 32,1 \text{ kNm, ou } e_x = 0,025\text{m}$$

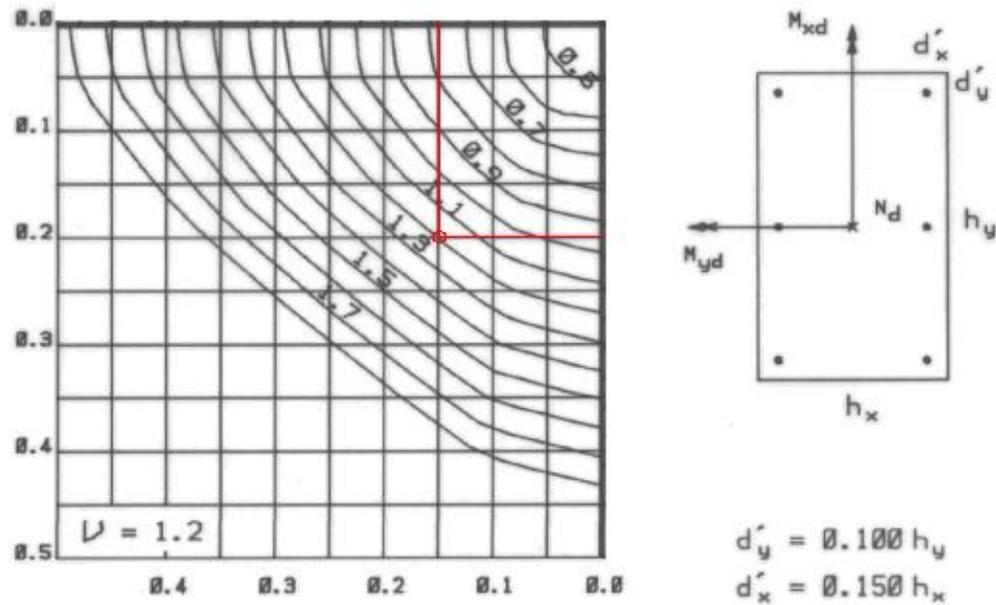
Lembrando que  $M_x$  e  $M_y$  desta bibliografia significam nas variáveis usadas no atual trabalho respectivamente,  $M_{yy}$  e  $M_{xx}$ . Para facilitar, apresenta-se o problema graficamente, na Figura 40 :

Figura 40 - Seção a ser dimensionada no exemplo 3.



Assim, usando as equações (4.1), (4.2) e (4.3), são obtidos os valores  $v = 1,2$ ,  $\mu_x = 0,15$  e  $\mu_y = 0,2$ . O ábaco que descreve tal situação é o abaco 29 na publicação de Pinheiro, Baraldi e Porem (2009). É apresentado parcialmente na Figura 41.

Figura 41 – Ábaco relativo ao exemplo 3.



Fonte: Pinheiro, Baraldi e Porem (2009).

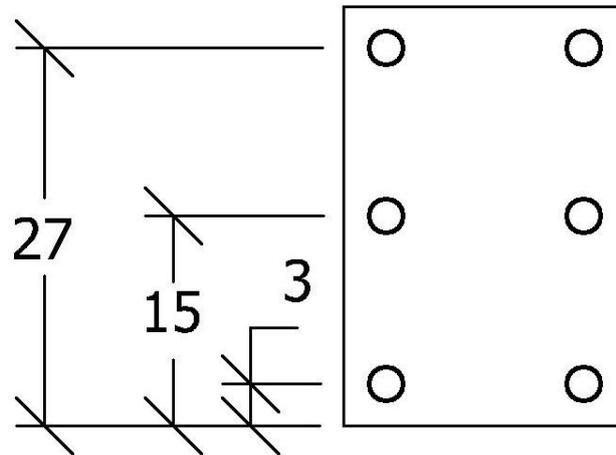
Ao se consultar o ábaco, chega-se à taxa mecânica de armadura  $\omega = 1,2$ . (4.4)

$$\omega = \frac{A_s \cdot f_{yd}}{A_c \cdot f_{cd}} \rightarrow A_s = \frac{\omega \cdot A_c \cdot f_{cd}}{f_{yd}}$$

$$A_s = \frac{\omega \cdot A_c \cdot f_{cd}}{f_{yd}} = \frac{1,2 \cdot 600 \cdot 1,7857}{43,478} = 29,57 \text{ cm}^2$$

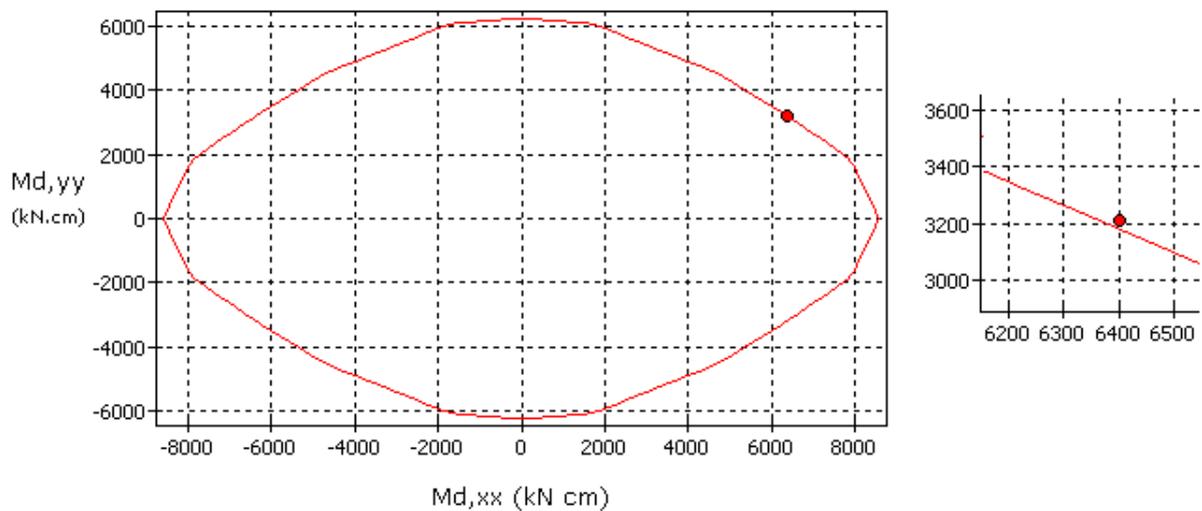
Portanto, a seção apresenta 6 barras cuja área é 4,928cm<sup>2</sup>. Assim, as medidas e a conformação da armadura (adotada seguindo os ábacos de Pinheiro, Baraldi e Porem (2009), que exigem conformação da armadura pré-definida) constam na Figura 42. A distância entre o centro da barra e as bordas da seção é 3cm.

Figura 42 – Seção dimensionada no exemplo 3 e posições da armadura.



A Figura 43 mostra os resultados do exemplo 3. Além disso, devido ao ponto estar muito próximo da reta que limita o diagrama de momentos resistentes, é apresentado um zoom na mesma imagem.

Figura 43 – Seção aplicada no programa, seus resultados e um zoom no mesmo gráfico.



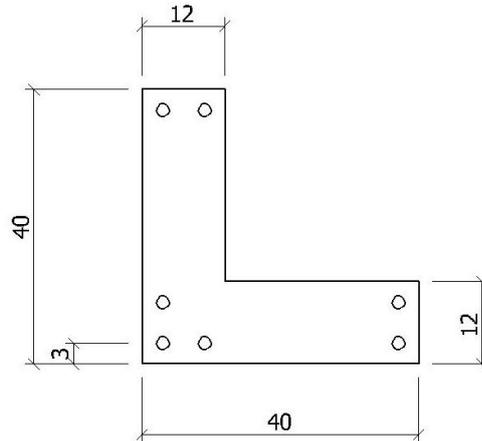
O comportamento obtido foi considerado satisfatório, pois o resultado do programa e do ábaco estão próximos, sujeitos apenas a pequenas diferenças, como devido à leitura do ábaco ou diferentes considerações de cálculo.

#### 4.4 EXEMPLO 4 – DIAGRAMA DE MOMENTOS RESISTENTES

Este exemplo tem como objetivo comparar diagramas de momentos resistentes com um exemplo da bibliografia Santos (1994).

A seção a ser avaliada tem forma de “L”, e suas medidas são mostradas na Figura 44. Os pontos do polígono da seção de concreto são: (0 , 0); (40 , 0); (40 , 12); (12 , 12); (12 , 40); (0 , 40). Os pontos da armadura passiva: (3 , 3) ; (3 , 9) ; (9 , 3) ; (37 , 3) ; (37 , 9) ; (3 , 37) ; (9 , 37). As áreas de cada uma é 2 cm<sup>2</sup>. O  $f_{ck}$  é 20MPa, o  $\gamma_c$  é 1,5, aço CA-50, com  $\gamma_s$  é 1,15.

Figura 44 – Medidas da seção do exemplo 4.



Santos (1994) apresenta diagramas de momento resistentes para  $N_d = 1000\text{kN}$  (compressão) e  $N_d = -500$  (tração). Ambas situações também foram avaliadas com as rotinas deste trabalho. Para a primeira situação de esforço normal os resultados constam na Figura 45

Já para a segunda situação de esforço normal, 500kN de tração, os resultados são apresentados na Figura 46.

Figura 45 – Diagrama de momentos resistentes para  $N_d = 1000\text{kN}$  (compressão)

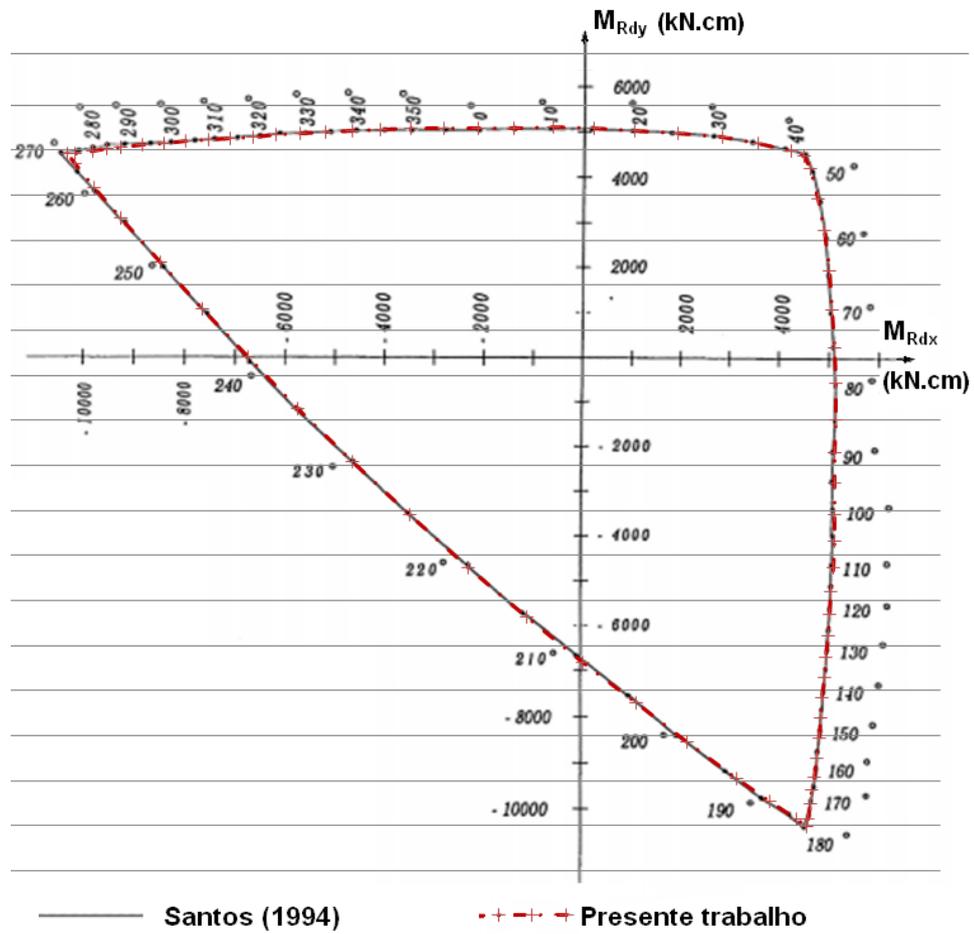
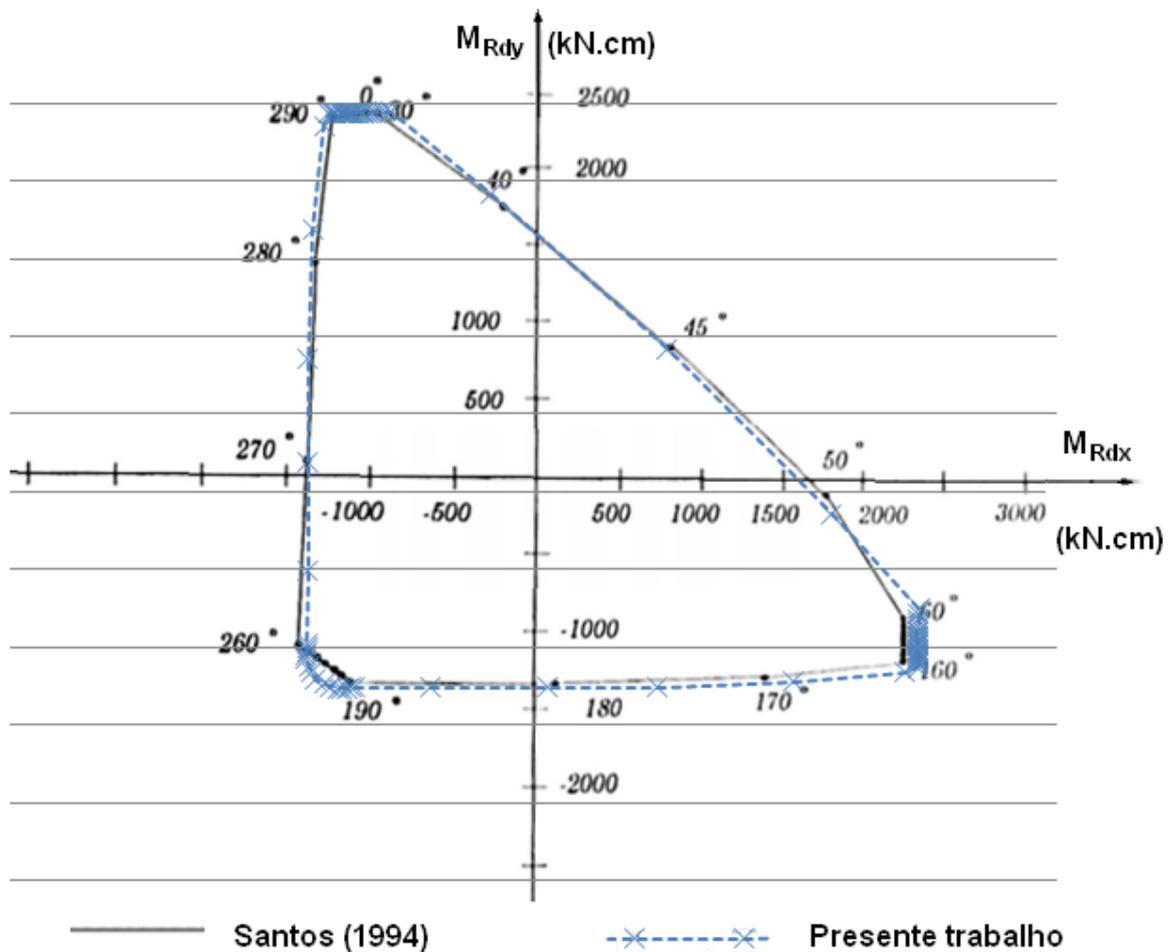


Figura 46 –Diagrama de momentos resistentes para  $N_d = -500$  kN (Tração).



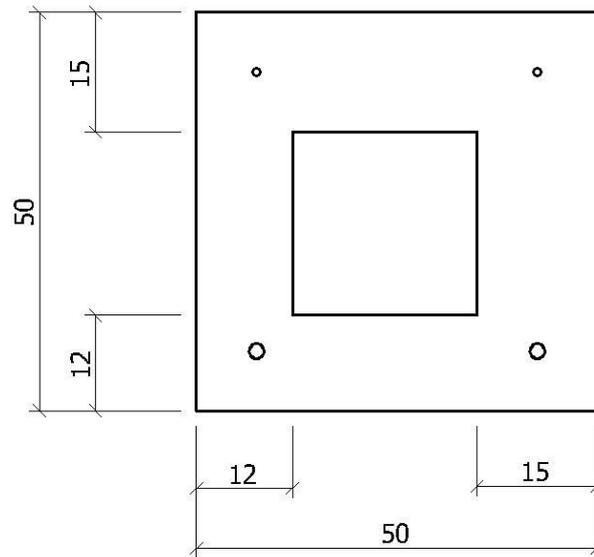
Em ambos gráficos, os momentos resistentes seguiram conformações semelhantes.

#### 4.5 EXEMPLO 5 – DIAGRAMA DE MOMENTOS RESISTENTES COM OBLIQUA 1.0.

Este exemplo apresenta uma seção vazada em estado-limite último, comparada com o programa Oblíqua 1.0 (2001), desenvolvido pela Universidade Federal do Paraná em 2001, um programa que calcula as envoltórias de momentos resistentes de seções submetidas à flexão oblíqua composta. O programa foi desenvolvido na Universidade Federal do Paraná e é uma referência neste tema.

A seção a ser testada é uma seção vazada, cuja seção de concreto é apresentada pela Figura 47. Os materiais são concreto C30 e aço CA-50. As armaduras positivas tem área unitária  $3,2\text{cm}^2$ . As armaduras negativas possuem área unitária  $0,8\text{cm}^2$ . A distância das barras de armadura até as bordas (cobrimento + metade do diâmetro) foi adotada como  $7,5$  cm – relativo apenas às bordas externas ignorando cobrimento em relação à região vazada.

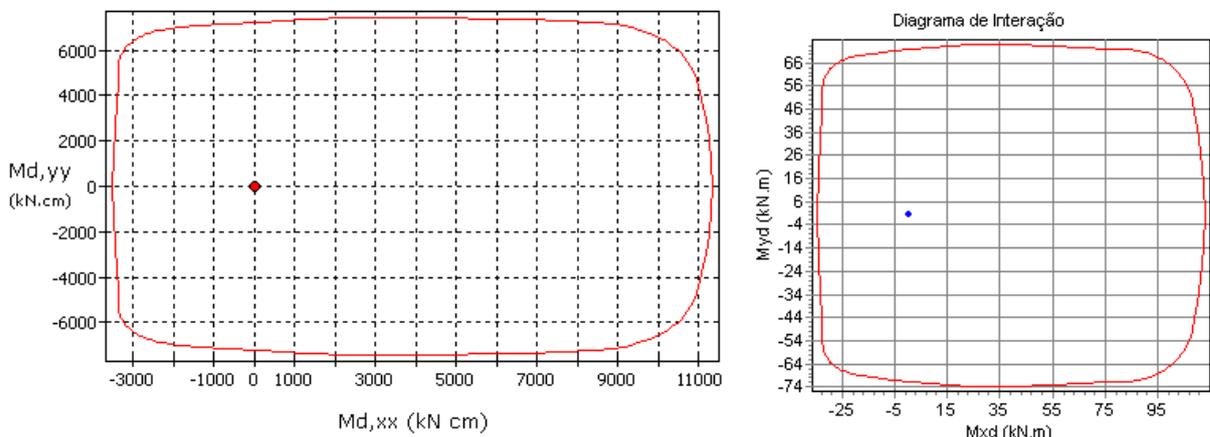
Figura 47 – Seção a ser analisada no exemplo 5, com cotas em cm.



No programa deste trabalho, são inseridos os pontos da poligonal de concreto, na seguinte ordem: (0 ; 0) , (50 ; 0) , (50 ; 35) , (35 ; 35) , (35 ; 12) , (12 ; 12) , (12 ; 35) , (35 ; 35) , (50 ; 35) , (50 ; 50) , (0 ; 50). Como já citado, para figuras vazadas, desenha-se a figura externa com pontos em sequência anti-horária, e as partes vazadas com uma sequência horária de pontos.

Os pontos que definem a posição da armadura seguem como dispostos na Figura 47, e são apresentados a seguir, na forma (*posição x, posição y, armadura*): (7,5 ; 7,5 ; 3,2cm<sup>2</sup>) , (42,5 ; 7,5 ; 3,2cm<sup>2</sup>) , (7,5 ; 42,5 ; 0,8cm<sup>2</sup>) , (42,5 ; 42,5 ; 0,8cm<sup>2</sup>) . Os resultados são apresentados na Figura 48: à esquerda os resultados do programa deste trabalho e à direita os resultados do Oblíqua 1.0. Em ambos gráficos, há uma marcação fixa no ponto de momento nulo ( $M_{xx} = 0; M_{yy} = 0$ ).

Figura 48 – Resultados segundo este trabalho à esquerda e do Oblíqua à direita.



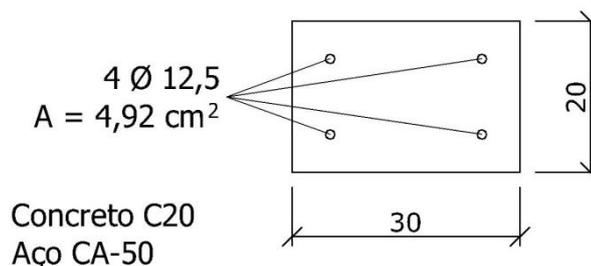
Embora a apresentação dos resultados seja feita em escalas diferentes, há semelhança visível nos valores e na conformação dos gráficos.

#### 4.6 EXEMPLO 6 – RELAÇÃO MOMENTO X CURVATURA USANDO PCALC

O exemplo 6 tem por objetivo avaliar os resultados de momento-normal-curvatura obtidos pelo programa e suas rotinas. Para isso, compara-se os resultados obtidos com as rotinas deste trabalho com os resultados do programa PCalc 1.1 (2013) – programa para análise não-linear de pilares de concreto armado. Mais informações sobre este programa estão disponíveis em seu endereço eletrônico, e também por Cardoso Júnior e Kimura (2013), disponíveis nas referências.

A seção avaliada apresenta as medidas: 30 cm de base, 20cm de altura, 4 barras de diâmetro de 12,5mm, com área unitária 1,23cm<sup>2</sup>. A distância  $d'$  até aborda é de 5cm. Os materiais são concreto C20 e aço CA-50, conforme apresentada na Figura 49. As relações momento-curvatura são apresentadas para as forças axiais de compressão de cálculo 280kN e 588kN.

Figura 49 – Seção do exemplo 6.



O programa PCalc apresenta 2 curvas para momento-curvatura em seu gráfico: uma para  $0,85 f_{cd}$ , e outra para  $1,1 f_{cd}$ . No intuito de gerar a mesma curva nas rotinas deste trabalho, pode se multiplicar a resistência por  $1,1/0,85 = 1,2941176$ , que resulta em 25,88MPa. Portanto, são calculadas 2 curvas para comparação, uma com  $f_{ck} = 20$  MPa, que determina a curva  $0,85 f_{cd}$ , e outra com  $f_{ck} = 25,88$ MPa, para determinar a curva  $1,10 f_{cd}$ .

Além disso, o programa PCalc utiliza valores característicos de esforços, e neste trabalho se calcula com esforços em seus valores de cálculo. Logo, para simular o mesmo carregamento, se no programa PCalc se supõe o carregamento a compressão de 200 kN ( $280 \text{ kN}/1,4$ ), neste trabalho se utiliza 280 kN. Já o valor de 420kN no PCalc é considerado por ( $588 \text{ kN}/1,4$ ) nos resultados deste trabalho.

Para a força característica de 200kN, os resultados segundo este trabalho e segundo o PCalc são mostrados na Figura 50.

Já ara a força característica de 420kN, são mostrados na Figura 51.

Figura 50 – Relações Momento-Curvatura sendo a força normal característica 200 kN.

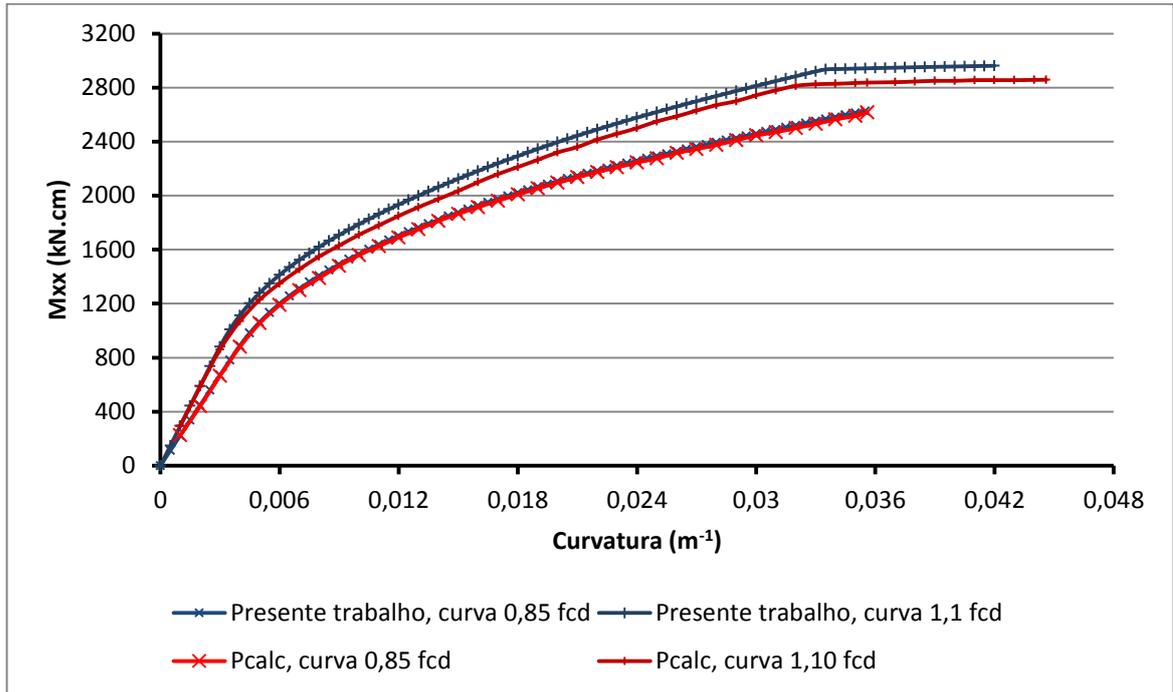
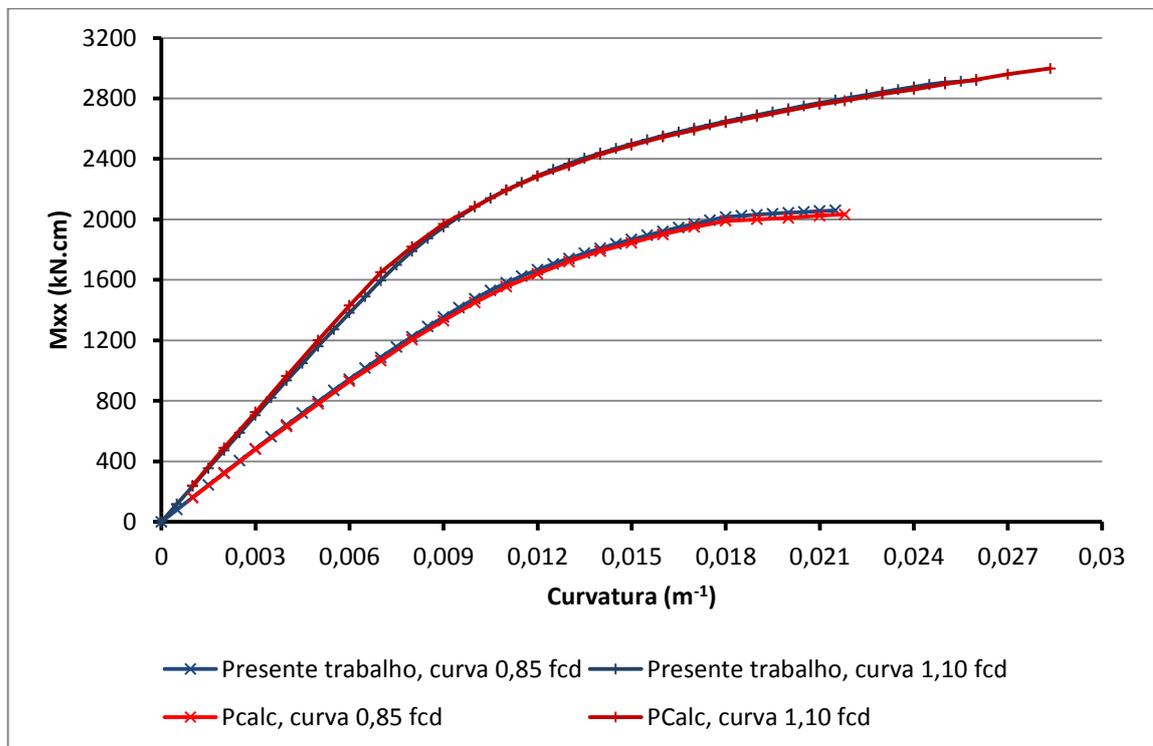


Figura 51 - Relações Momento-Curvatura sendo a força normal característica 420 kN



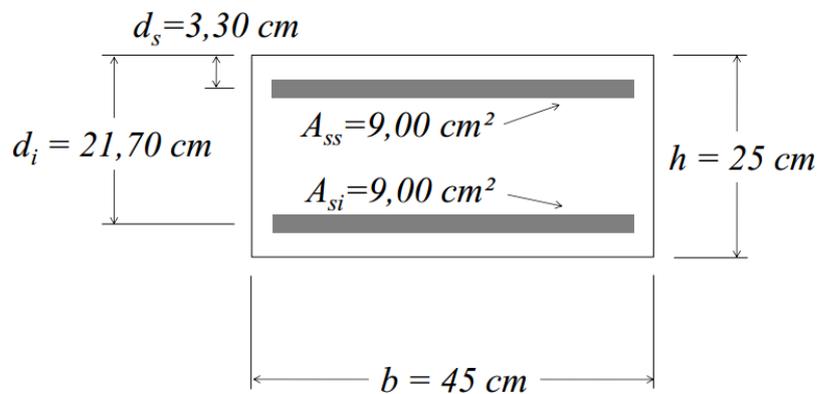
As relações são extremamente próximas em ambos os gráficos. Há alguma diferença numérica devido ao método e às considerações, que podem divergir em vários pontos entre os programas.

#### 4.7 EXEMPLO 7 – GRÁFICO MOMENTO X NORMAL X CURVATURA COM RIBEIRO (2011)

Este exemplo tem por objetivo comparar os gráficos momento-normal-curvatura criados pelo programa desenvolvido na dissertação e outra fonte. Para isso se apresenta um exemplo de curva momento-curvatura descrito por Ribeiro (2011). Em seu trabalho o autor apresenta o gráfico momento-curvatura em equações. Isso é importante, pois possibilita comparar os valores.

A relação momento-curvatura é referente à seção da Figura 52, com concreto  $f_{ck} = 20MPa$ , aço com  $f_{yk} = 500 MPa$ , como descreve o autor.

Figura 52 - Seção Transversal do Exemplo 4.



Fonte: Ribeiro (2011).

Porém, as equações do autor são referentes à curvatura adimensional  $\theta$ , cuja equação é dada por (4.5). Além disso, são elaborados vários gráficos para a seção, mas sob diferentes forças normais. As forças normais também são apresentadas em relação ao esforço normal adimensional  $\nu$ , expresso em (4.6), segundo Ribeiro (2011).

$$\theta = 1000 \cdot h \cdot \frac{1}{r} \quad (4.5)$$

$$\nu = \frac{N_d}{A_c \cdot f_{cd}} \quad (4.6)$$

Em sua publicação, o autor apresenta equações para diversas condições. Para o presente trabalho são apresentadas três comparações: uma para cada um dos seguintes valores de esforço normal adimensional:  $\nu = 0$ ;  $\nu = 0,2$  e  $\nu = 0,5$ . Além disso, Ribeiro (2011) separa a função em dois trechos quando necessário, para aproximar melhor o comportamento das funções.

As equações originais são apresentadas Tabela 3 e são divididas em dois trechos quando necessário. Existe um intervalo em que se deve usar as equações, indicado na coluna *Validade* da tabela. Numa planilha paralela são calculados vários pontos em sequencia, para

cada esforço normal adimensional, a fim de se gerar um gráfico com esses pontos. Os valores exatos não são apresentados, pois podem ser calculados diretamente.

Finalmente, para obter os valores do programa, se aplicam os valores análogos de esforços normais adimensionais no programa ( $v=0,0 \rightarrow N_d = 0$ ;  $v=0,2 \rightarrow N_d = 321,43$  kN;  $v=0,5 \rightarrow N_d = 803,57$  kN) na seção proposta inicialmente do exemplo. Os resultados obtidos para a curvatura são apresentados em termos de  $m^{-1}$ , logo, é necessário converter estes valores para curvatura adimensional usando a equação (4.5). Os valores resultantes para momento-curvatura constam na Tabela 4. Em caso de houver quaisquer dúvidas sobre exportação dos dados do programa, sugere-se consultar a seção 5.3.

Tabela 3 – Equações segundo a bibliografia para o exemplo 7.

		Trecho curvo – Equação Polinomial			Trecho Reto – Equação linear	
$v$	$M_u$ (kN)	Equação	$r^2$	Validade	Equação	Validade
0,0	74,73	$M = -0,0569 \cdot \theta^3 - 0,0053 \cdot \theta^2 + 20,355 \cdot \theta$	1	$0 \leq \theta \leq 3,8$	$M = 0,2098 \cdot \theta + 73,353$	$3,8 \leq \theta \leq 6,56$
0,2	101,97	$M = -0,059 \cdot \theta^6 + 0,9559 \cdot \theta^5 - 6,2447 \cdot \theta^4 + 21,004 \cdot \theta^3 - 21,736 \cdot \theta^2 + 54,122 \cdot \theta$	1	$0 \leq \theta \leq 4,6$	$M = 0,8739 \cdot \theta + 95,125$	$4,6 \leq \theta \leq 7,83$
0,5	102,3	$M = 0,1546 \cdot \theta^6 - 2,1273 \cdot \theta^5 + 11,102 \cdot \theta^4 - 26,088 \cdot \theta^3 + 20,195 \cdot \theta^2 + 37,642 \cdot \theta$	1	$0 \leq \theta \leq 4,08$	-	-

Fonte: Ribeiro (2011).

Tabela 4 – Resultados originados pelo programa para apresentação gráfica.

$v = 0,0$		$v = 0,2$		$v = 0,5$	
$\theta$	M (kN)	$\theta$	M (kN)	$\theta$	M (kN)
0	0	0	0	0	0,00
1	19,36	1	30,79	0,3	10,18
2	38,23	2	49,87	0,6	20,32
3	56,55	3	67,34	0,9	30,36
4	72,74	4	83,77	1,2	40,04
5	73,35	4,5	91,57	1,5	48,00
6	73,75	5	97,75	1,8	54,78
7	74,03	5,5	98,42	2,1	60,82
7,5	74,14	6	98,98	2,4	66,36
8	74,23	6,5	99,46	2,7	71,53
8,5	74,31	7	99,87	3	76,39
9	74,38	7,5	100,23	3,3	81,01
9,5	74,43	8	100,55	3,6	85,44
10	74,48	8,5	100,83	3,9	89,74
10,5	74,52	9	101,08	4,2	93,95
11	74,56	9,5	101,31	4,5	96,73
11,5	74,59	10	101,51	4,8	98,34
12	74,62	10,5	101,69	5,1	99,82
12,5	74,64	11	101,86	5,4	101,17
12,6	74,65	11,4	101,98	5,6	102,01

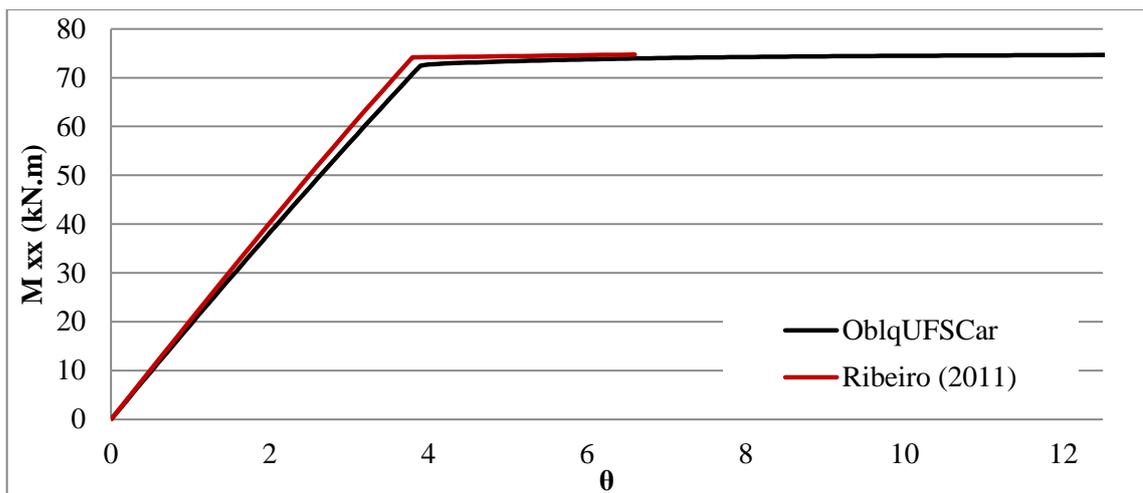
Figura 53 – Gráfico Momento – Curvatura adimensional para  $v = 0,0$ .

Figura 54 – Gráfico Momento – Curvatura adimensional para  $v = 0,2$ .

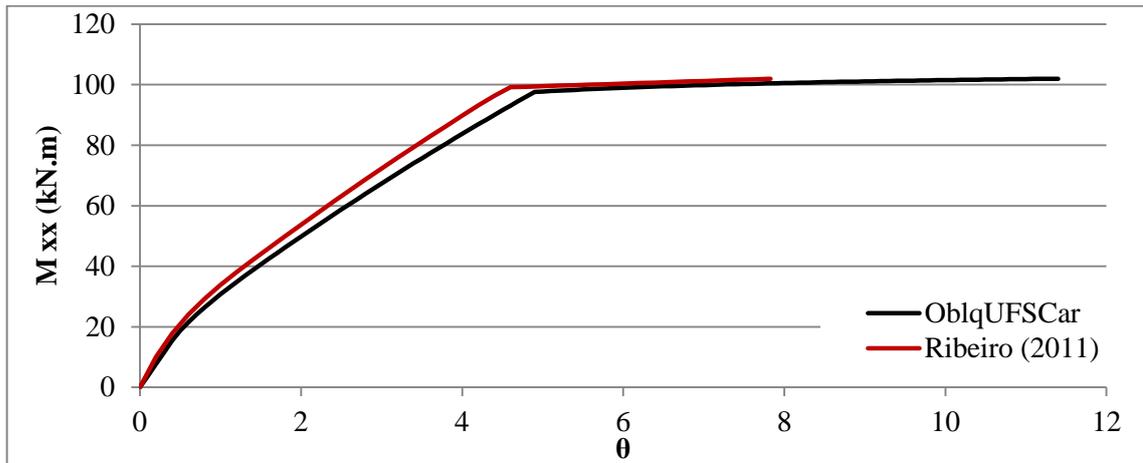
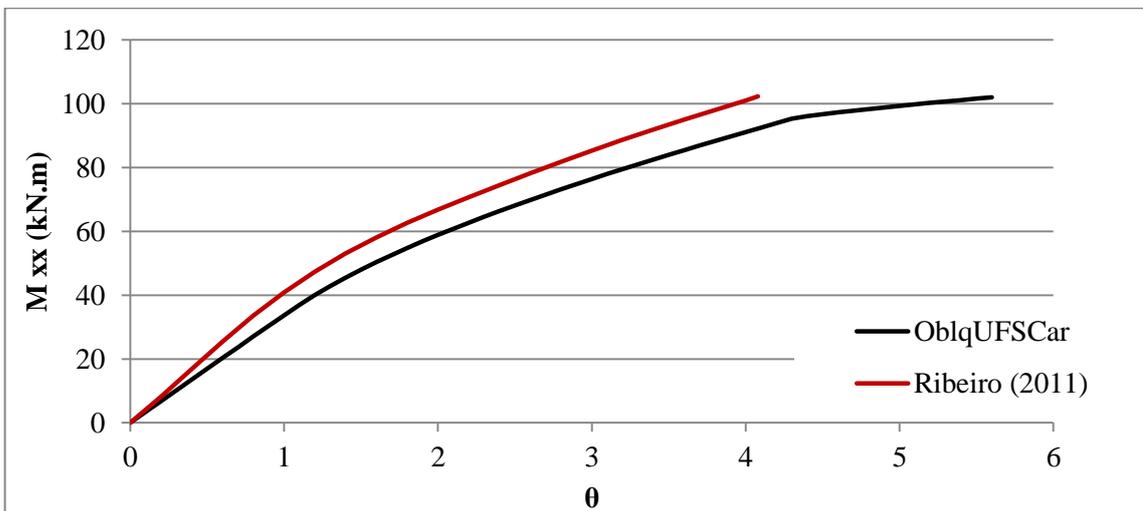


Figura 55 – Gráfico Momento – Curvatura adimensional para  $v = 0,5$ .



#### 4.8 EXEMPLO 8 – VERIFICAÇÕES DE SERVIÇO

Nesta seção apresenta-se a verificação do estado-limite de serviço de uma seção em concreto protendido: um exemplo apresentado por Carvalho (2012). Trata-se do exemplo numérico 7.1 da obra. Nele, o autor apresenta detalhadamente as verificações de serviço de uma laje maciça, de 10x10 m, altura de 25cm e protendida por cordoalhas engraxadas nas duas direções. Para realizar os cálculos, se considera a atuação da protensão convertendo-a numa carga distribuída equivalente  $u$ , cujos métodos são explicados na bibliografia.

Conhecendo-se as cargas distribuídas, o autor calcula os momentos máximos usando a teoria de placas delgadas, para isso consultando a tabela 7.3, do capítulo 7 da obra Carvalho (2012).

- carga permanente (peso próprio, sobrecarga permanente): 6 kN/m<sup>2</sup>;
- carga acidental, 5 kN/m<sup>2</sup>;
- efeito da protensão, -4 kN/m<sup>2</sup>;

Considerando os carregamentos da laje, e usando os valores tabela dos, pode-se calcular os seguintes momentos máximos atuantes na laje devido aos esforços citados anteriormente:

- carga permanente: 35,28 kN.m/m = 3528 kN cm/m;
- carga acidental: 22,0 kN.m/m = 2200 kN cm/m;
- efeito da protensão: -17,64 kN.m/m = 1764 kN cm/m;

Então, são necessárias as verificações de estado-limite de serviço de formação de fissuras (combinação frequente  $\Psi = 0,4$ ) e de descompressão (combinação quase permanentes  $\Psi = 0,3$ ).

Para combinação frequente, tem-se como momentos máximo e mínimo:

$$M_{max} = 3528 + 0,4 \cdot 2200 - 1764 = 2644 \text{ kN} \cdot \text{cm/m}$$

$$M_{min} = 3528 + 0 \cdot 2200 - 1764 = 1764 \text{ kN} \cdot \text{cm/m}$$

Para combinação quase permanente, tem-se como momentos máximo e mínimo:

$$M_{max} = 3528 + 0,3 \cdot 2200 - 1764 = 2424 \text{ kN} \cdot \text{cm/m}$$

$$M_{min} = 3528 + 0 \cdot 2200 - 1764 = 1764 \text{ kN} \cdot \text{cm/m}$$

Para inserir os dados no programa, primeiro recomenda-se abrir a janela de materiais e alterar a resistência do concreto para 35 MPa. Então, introduz-se os dados da seção: como o efeito da protensão foi convertido em momento e força normal, não se insere armadura ativa. Além disso, como não foi citada armadura passiva, não se insere nenhuma armadura passiva também. A seção inserida deve ter base de 1m e altura de 25cm, e os pontos da poligonal podem ser quaisquer, desde que descrevam essa conformação. A Figura 56 apresenta uma poligonal para exemplificar.

Figura 56 – Poligonal do exemplo 8.

Número de pontos: 4		
	X (cm)	Y (cm)
1	-50	0
2	50	0
3	50	25
4	-50	25

Ao se acionar o botão de “Calcular”, o programa retorna os resultados: ângulo e deformação nos pontos extremos da seção. Além disso, o programa apresenta as verificações de serviço: ELS-W, ELS-F e ELS-D.

Logo, há 4 valores de solicitações a serem testados: a força axial é de 370 kN para todos. Os momentos são ELS-F: 2644 kN.cm, 1764 kN.cm, ELS-D: 2424 kN.cm, 1764 kN.cm. Como um dos valores se repete, apenas 3 iterações são realizadas. Os resultados são apresentados na Tabela 5.

Tabela 5. Resultados das verificações do exemplo 8.

	Situação 1; $M_{xx} = 2644 \text{ kN.cm}$	Situação 2 e 4; $M_{xx} = 1764 \text{ kN.cm}$	Situação 3; $M_{xx} = 2424 \text{ kN.cm}$
$\varepsilon_1$ (‰)	0,22745	0,15509	0,20367
$\varepsilon_2$ (‰)	-0,12984	-0,013117	-0,084022
A	0	0	0
Nro. de iterações	200	68	145
Menor deform.	-0,12984‰	-0,013117‰	-0,084022‰
ELS-W	Impossível, não há armadura na seção		
ELS-F*	<b>Atende</b>	<b>Atende</b>	Atende
ELS-D**	<b>Não atende</b>	<b>Não atende</b>	<b>Não atende</b>
*Lembrando que o programa compara a menor deformação que ocorre na seção de concreto com -0,15‰, ver seção 2.6.3 e Figura 18.			
**Para avaliar, compara-se a menor deformação que ocorre na seção de concreto com zero, ver seção 2.6.4			

Assim, os resultados finais são os mesmos obtidos por Carvalho (2012), isto é, atende-se ao ELS-F (tanto momento máximo quanto mínimo), mas não se atende ao ELS-D (tanto momento máximo quanto mínimo, novamente).

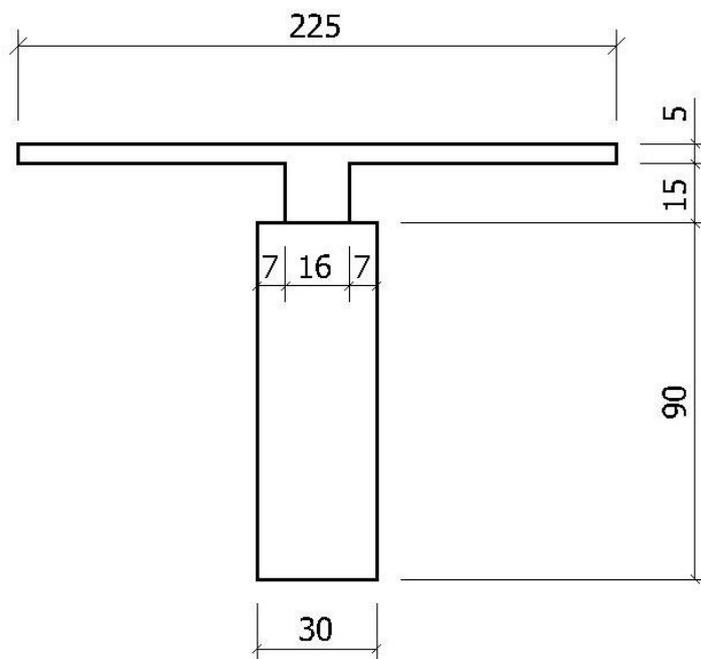
#### 4.9 EXEMPLO 9 – VERIFICAÇÃO DE SERVIÇO DE ABERTURA DE FISSURAS COM PROTENSÃO

Para avaliar este exemplo, foi tomado um exemplo apresentado por Faleiros Junior (2010), em que o autor apresenta a verificação de abertura de fissuras que ocorre na armadura ativa.

Na obra referenciada, o autor apresenta detalhadamente a origem de cada solicitação e o cálculo do dimensionamento, e em caso de interesse, trata-se do caso *b* do exemplo 10, apresentado na seção 7.10 daquela bibliografia.

O autor calcula a armadura passiva e ativa da seção composta descrita pela Figura 57. Nesta, a parte superior (capa) possui concreto C30, e a parte inferior (viga), C40. O momento atuante na combinação de ações considerada é de 672,7 kN.m ou 67270 kN.cm.

Figura 57 – Seção referente ao exemplo 9



Além disso, a disposição das armaduras não é apresentada na figura, e a armadura passiva não é comercial, mas apresentada como o valor exato obtido no cálculo. Para armadura passiva positiva tem-se  $A_s = 7,05 \text{ cm}^2$ ,  $d = 105 \text{ cm}$ . Para armadura passiva negativa, tem-se  $A_s' = 4,56 \text{ cm}^2$  com  $d = 5 \text{ cm}$ . Sendo  $d$  a distância do centro da armadura ao topo da seção.

Para armadura ativa, utiliza-se 4 cordoalhas de 12,7mm ( $\Phi = 1/2''$ , com área de aço transversal de  $0,98 \text{ cm}^2$ ) de aço CP 190RB, que após as perdas apresenta  $\sigma_{p,t=\infty} = 1119 \text{ MPa}$ .

Assim, tem-se o necessário para a verificação: uma seção definida: com armaduras passivas e ativas também definidas, além das características dos materiais.

Primeiramente, resolve-se o problema de se avaliar uma seção composta de dois materiais diferentes, para isso, faz-se a homogeneização da seção. Faleiros Junior (2010) calcula uma seção homogeneizada em comparação aos módulos de elasticidade do concreto, que resulta numa razão apresentada pela equação (4.7).

$$r = \frac{E_{capa}}{E_{viga}} = \frac{0,85 \cdot 5600 \cdot \sqrt{30}}{0,85 \cdot 5600 \cdot \sqrt{40}} = 0,866 \quad (4.7)$$

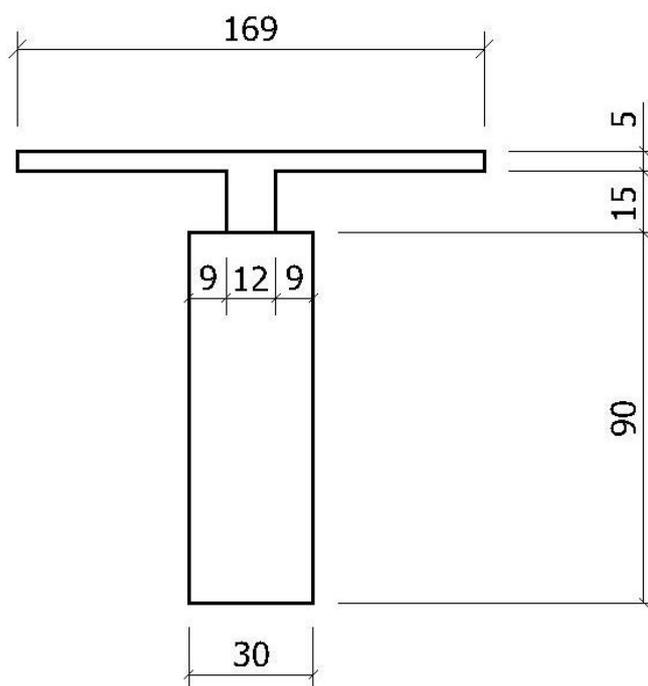
Porém, o presente trabalho não trabalha com essa abordagem: por isso a redução segue outra razão, apresentada pela equação (4.8). Salienta-se também que caso fosse usada a

abordagem de se considerar o módulo de elasticidade  $E$  como sendo a derivada da tensão pela deformação, ainda assim se obtém o mesmo resultado.

$$r = \frac{0,85 \cdot \frac{30}{1,4} \cdot \left[1 - \left(1 - \frac{\varepsilon_c}{\varepsilon_{c2}}\right)^2\right]}{0,85 \cdot \frac{40}{1,4} \cdot \left[1 - \left(1 - \frac{\varepsilon_c}{\varepsilon_{c2}}\right)^2\right]} = 0,75 \quad (4.8)$$

Portanto, a seção apresentada pela Figura 57 é transformada, e agora possui a conformação descrita pela Figura 58, e a composição é inteira com concreto C40.

Figura 58 – Seção do exemplo 9 homogeneizada



Primeiramente, altera-se no programa os dados dos materiais: e altera-se o valor de  $f_{ck}$  30MPa para 40 MPa. Então, são inseridos os pontos que definem a poligonal da seção concreto: (0 ; 0) ; (30 ; 0) ; (30 ; 90) ; (21 ; 90) ; (21 ; 105) ; (99,5 ; 105) ; (99,5 ; 110) ; (-69,5 ; 110) ; (-69,5 ; 105) ; (9 ; 105) ; (9 ; 90) ; (0 ; 90).

Então, são dispostos no programa os dados das armaduras, realizado de maneira a forçar a seção à ser simétrica em relação ao eixo  $y$ . Isso deve ser feito sempre que se analisa uma seção em que o momento em torno do eixo  $y$  não for importante, que é o caso.

Os valores inseridos seguem os dados já apresentados, com exceção da deformação de pré-alongamento da armadura ativa, que deve ser obtido calculando as perdas totais da armadura. Para calcular o valor da deformação de pré-alongamento, parte-se da tensão citada, isto é, 1119 MPa: utiliza-se a equação (2.9) ou (2.10) (dependendo da região, ver Figura 11), e obtém-se o valor de 5,595‰, que é inserido no programa como mostra a Figura 59. A deformação é de tração, mas sua inserção é positiva, para não confundir na inserção de dados,

porém, no sistema interno do programa é reservado com sinal negativo, para simbolizar a tração adequadamente.

Finalmente, apresenta-se os dados aplicados no programa para a armadura passiva e ativa, expostos na Figura 59. Alguns pontos também são apresentados, para apresentar como a interface do programa diante da presente verificação.

Figura 59 – Dados geométricos da seção do exemplo 5.

The screenshot shows a software interface with the following components:

- Materiais** (Materials) tab selected.
- Checkboxes for  Armadura Passiva and  Armadura Ativa.
- Input fields:
  - Número de pontos: 12
  - Número Barras: 2
  - Número Barras: 4
- Table 1 (Points):
 

	X (cm)	Y (cm)
5	21	105
6	99,5	105
7	99,5	110
8	-69,5	110
9	-69,5	105
10	9	105
11	9	90
12	0	90
- Table 2 (Reinforcement Bars):
 

	x (cm)	y (cm)	As (cm <sup>2</sup> )
1	15	5	7,05
2	15	105	4,56
- Table 3 (Reinforcement Bars with Properties):
 

	x (cm)	y (cm)	As (cm <sup>2</sup> )	ε <sub>p</sub> (‰)	Φ (mm)
1	6	5	0,98	5,595	12,7
2	12	5	0,98	5,595	12,7
3	18	5	0,98	5,595	12,7
4	24	5	0,98	5,595	12,7

Após o cálculo, o programa retorna o resultado:  $\varepsilon_1 = 0,32394\text{‰}$ ;  $\varepsilon_2 = -1,0626\text{‰}$  e  $\alpha = 0^\circ$ . Retorna também os valores de abertura de fissuras para a armadura passiva, que embora não sejam descritos como bitolas comerciais, ainda são citados: da armadura passiva positiva:  $0,191092\text{ mm}$ , e da armadura passiva negativa  $0\text{ mm}$ .

Já para a armadura ativa, todas as quatro cordoalhas apresentam a mesma tensão de descompressão:  $-199,91\text{ MPa}$ , e a abertura de fissura  $w_k = 0,1446\text{ mm}$ . A tensão poderia ser calculada a partir de que se conhece a conformação de deformação na seção. Por exemplo: considerando  $\sigma$  como uma função de  $\varepsilon_t$ , e  $\varepsilon_t = \varepsilon_p + \varepsilon_s$  a tensão a ser considerada  $\sigma_{si}$  é a tensão sob atuação das solicitações avaliadas menos a tensão em descompressão (ver último parágrafo da seção 3.7.2).

Em serviço:  $\varepsilon_s = -0,999588$ ; Em descompressão:  $\varepsilon_s = 0$ ; Em serviço e em descompressão,  $\varepsilon_s = -5,595$ . Logo, lembrando a convenção de sinais (negativo = tração; alargamento), tem-se o exposto em (4.9), apresentando como foi obtido o resultado do programa:

$$\begin{aligned}\sigma_a \text{ ser considerada} &= \sigma_{pi,w} = \sigma_{serviço} - \sigma_{descompressão} \\ \sigma_{pi,w} &= \sigma(\varepsilon_t = -0,99958 - 5,595) - \sigma(\varepsilon_t = -0,99958 - 5,595) \\ \sigma_{pi,w} &= \sigma(\varepsilon_t = -6,59458) - \sigma_{descompressão}(\varepsilon_t = -5,595) \\ \sigma_{pi,w} &= -1318,916 - (-1119) = -199,916 \text{ MPa}\end{aligned}\tag{4.9}$$

A mesma análise segundo o autor Faleiros Junior (2010) tem resultado 0,4mm. A divergência ocorre devida à diferença entre as tensões, principalmente. A seguir consta a aplicação por Faleiros Junior (2010) na equação (4.10) e segundo a análise deste trabalho em (4.11), calculada automaticamente pelo programa.

$$w_k = \frac{\phi_i}{12,5\eta_1} \cdot \frac{\sigma_{si}}{E_{si}} \cdot \frac{3\sigma_{si}}{f_{ctm}} = \frac{12,5}{12,5 \cdot 1,2} \cdot \frac{342,65}{210000} \cdot \frac{3 \cdot 342,65}{0,3 \cdot 40^{\frac{2}{3}}} = 0,4mm\tag{4.10}$$

$$w_k = \frac{\phi_i}{12,5\eta_1} \cdot \frac{\sigma_{si}}{E_{si}} \cdot \frac{3\sigma_{si}}{f_{ctm}} = \frac{12,7}{12,5 \cdot 1,2} \cdot \frac{199,91}{200000} \cdot \frac{3 \cdot 199,91}{0,3 \cdot 40^{\frac{2}{3}}} = 0,1415mm\tag{4.11}$$

O autor obtém o valor de 0,2mm na segunda equação referente ao cálculo da fissuração (levando em conta características geométricas) – que não é calculada neste trabalho, pois as características geométricas usadas na segunda equação não são avaliadas pelo programa.

#### 4.10 EXEMPLO 10 – FLEXÃO RETA EM CONCRETO C90

O intuito deste exemplo é avaliar os resultados para uma seção com concreto de resistência à compressão característica de 90 MPa. Tal exemplo é importante pois ao se alterar a resistência característica à compressão do concreto, deve-se também alterar algumas outras características. Neste estudo, as variáveis que se alteram são as deformações  $\varepsilon_{c2}$  e  $\varepsilon_{cu}$ , e a variável  $n$ , utilizada na equação tensão-deformação do concreto, como é apresentada na Figura 9.

Para calcular a armadura, segue-se a orientação das equações apresentadas na palestra de Santos (2013). Calcula-se o KMD, a partir de (4.12), usando os dados conhecidos da seção. Então, consulta-se tabelas que fornecem os valores de  $kx$  e  $kz$  a partir de  $KMD$ . O valor de  $kz$  é usado então para calcular a área de armadura, usando a equação (4.13)

$$KMD = \frac{M_d}{b_w \cdot d^2 \cdot f_{cd}}\tag{4.12}$$

$$A_s = \frac{M_d}{k_z \cdot d \cdot \sigma_{sd}} \quad (4.13)$$

Considerando os dados do exemplo 1:  $M_d = 17,08$  kNm,  $b_w = 12$  cm;  $d = 29$  cm e  $f_{ck} = 90$  MPa, obtém-se  $KMD = 0,00263$ . Interpolando linearmente na Tabela 6, obtém-se  $k_z = 0,98$  e a área de aço resultante é de  $1,39$  cm<sup>2</sup>.

Tabela 6 – Parte da Tabela de dimensionamento de seções retangulares

$f_{ck}$ 90MPa	
KMD	$k_z$
0,009	0,993
0,019	0,986
0,028	0,979
0,037	0,972

Fonte: Santos (2013).

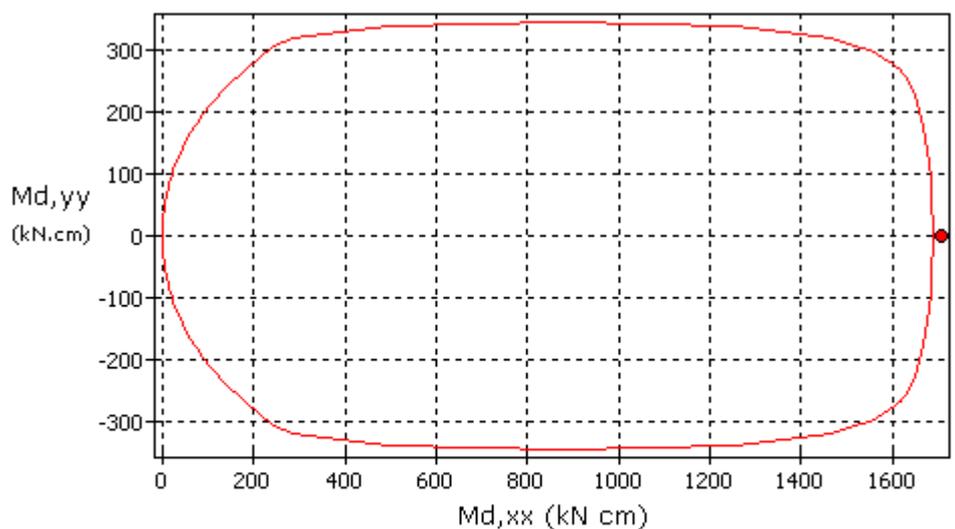
A área calculada foi para o momento fletor de  $17,08$  kNm, logo, para avaliar os resultados do programa, introduz-se nos dados da seção a armadura calculada, para se verificar qual o momento fletor o programa atribui ao ELU da seção sob flexão reta.

Constam a seguir os dados aplicados no programa:

- $f_{ck} = 90$  MPa;
- Poligonal da seção de concreto: (-6,-20); ( 6,-20) ; ( 6, 20) ; (-6, 20).
- Armadura passiva: Coordenada (0,-20), armadura: 1,39.

Os resultados constam na Figura 60. Nela, também constam os dados da barra de status do programa: o  $f_{ck}$ , o  $f_{cd}$  e o  $\varepsilon_{c2}$  e  $\varepsilon_{cu}$ .

Figura 60 – Resultados do exemplo 10



Angulo=0; Eps1=1,1891; Eps2=-10; x/h=0,10628; força=-4,9488E-6; Mxx = 1689,4; Myy = 0  
 $f_{ck} = 90$  MPa  $f_{cd} = 6,4286$  kN/cm<sup>2</sup>  $n = 1,4$  Deformações c2 e cu = 2,6 e 2,6

A variação entre os momentos é de 1,1%, relativamente pequena.

#### 4.11 EXEMPLO 11 – ESTADO-LIMITE ÚLTIMO DE ARMADURA ATIVA

O exemplo 11 tem por objetivo apresentar os resultados do cálculo para seções com armadura ativa.

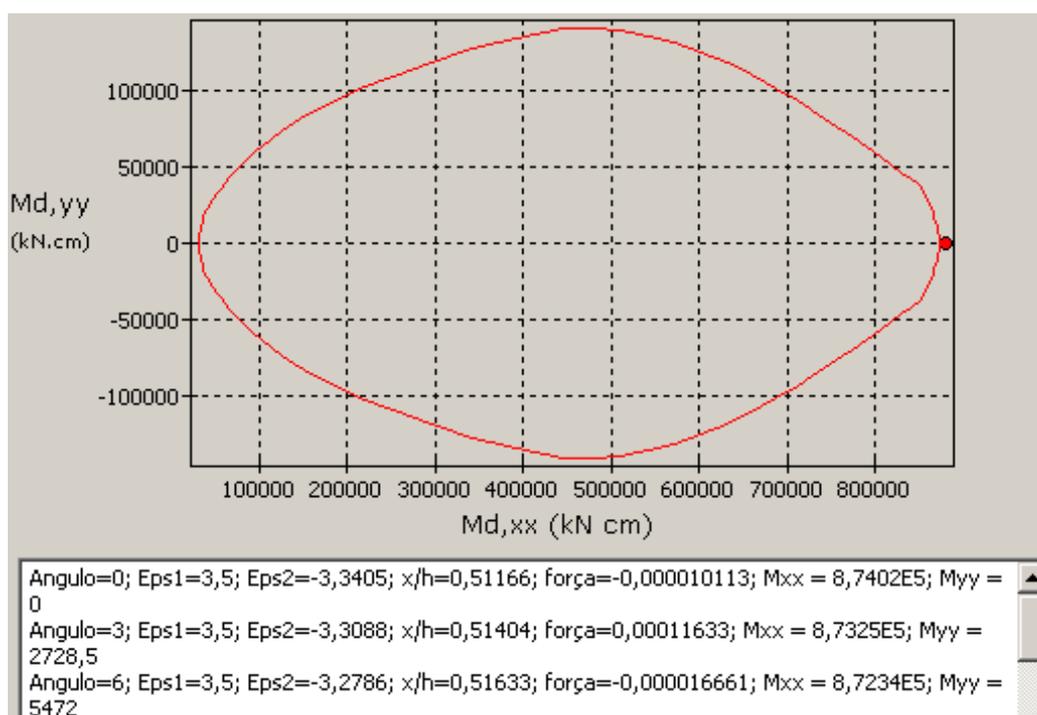
Trata-se de uma seção retangular, o exemplo 6.1 apresentado por Carvalho (2012). No exemplo, calcula-se a área de armadura ativa necessária à seção de 70cm de base, 145cm de altura útil e concreto C30 para resistir à um momento de cálculo de 8799 kN.m. A armadura utilizada é CP190 RB, e tensão no tempo infinito  $\sigma_{p\infty} = 1000$  MPa. Segundo o autor, a deformação de pré-alongamento é de 5,12‰, logo,  $E_p = 195$  GPa. As outras características da armadura ativa padrão são mantidas:  $f_{pyd} = 1460$  MPa,  $f_{ptd} = 1626$  MPa,  $\varepsilon_u = 35\%$ .

Então, todas as características são inseridas no programa, com exceção do momento fletor, cujo valor é obtido pela análise de estado-limite último.

Logo, os pontos da poligonal que define a seção de concreto: ( 0 , 0 ); ( 70 , 0 ); ( 70 , 145 ); ( 0 , 145 ); Não se inclui nenhuma armadura passiva: A armadura ativa se encontra no ponto ( 35 , 0 ), sua área é 52,2 cm<sup>2</sup>. A deformação de pré-alongamento é de 5,12‰, e no campo referente à bitola não é necessário alterar o valor padrão (zero), pois este campo é utilizado apenas para calcular o ELS-W.

Aplicando a rotina, o programa gera a envoltória de momentos, e o programa exporta os dados como texto também, como mostra a Figura 61.

Figura 61 – Parte dos resultados do exemplo 11.



O resultado obtido pelo programa é dado por: “ $Angulo=0$ ;  $Eps1=3,5$ ;  $Eps2=-3,3405$ ;  $x/h=0,51166$ ;  $força=-0,000010113$ ;  $M_{xx} = 8,7402E5$ ;  $M_{yy} = 0$ ”, usando exatamente a linguagem de saída. Logo, o momento fletor na ruptura é de 874020 kN.cm ou 8740,2 kN.m.

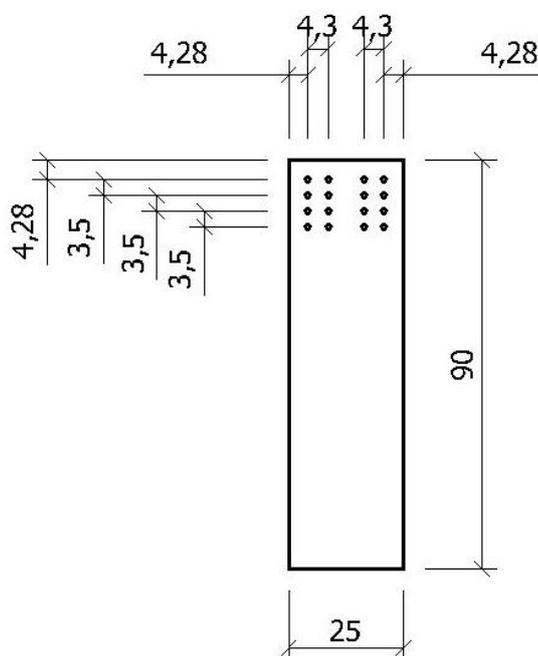
Comparando com o valor de momento fletor usado no dimensionamento, que é de 8799, há variação de 0,007, ou 0,7%.

#### 4.12 EXEMPLO 12 – ESTADO-LIMITE DE ABERTURA DE FISSURAS

Este exemplo tem o intuito de apresentar a verificação de abertura de fissuras numa seção em concreto armado, tendo como referência o exemplo 1 do capítulo 4 do livro de Estruturas de Concreto Armado, apresentado na referência Carvalho e Figueiredo Filho (2007).

A seção analisada é apresentada na Figura 62: uma seção de base 25cm, altura 90cm, concreto C20, armadura passiva CA-50. Trata-se de uma seção no apoio: cujo momento atuante negativo, considerado nesta verificação, é de 322,56 kN.m. As armaduras tem área unitária de 1,25cm<sup>2</sup>, e a área total é de 20cm<sup>2</sup>.

Figura 62 – Seção estudada no exemplo 12.



Porém, a análise apresentada no exemplo consta na bibliografia considerando o módulo de elasticidade como proposto pela seção 8.2.8 da norma ABNT NBR 6118:2014, apresentada pela equação (4.14). Este não é o tratamento por este trabalho, que considera as tensões no concreto como mostrado pela relação tensão-deformação no concreto, apresentado pela Figura 9.

$$E_{cs} = 0,85 \cdot 5600 \cdot \sqrt{f_{ck}} \quad (4.14)$$

Porém, a relação tensão-deformação usada neste trabalho é quase linear em seu início (como observado na Figura 9). Para comparar a análise do programa com a análise da referência, é necessário adotar módulos de elasticidades iguais (para que a distribuição de tensões seja semelhante). Então, apresenta-se uma manipulação da equação que determina a relação tensão-deformação no concreto recapitulada na equação (4.15), necessária para que o programa considere o módulo de elasticidade desejado.

$$\sigma_c = 0,85 \cdot f_{cd} \cdot \left[ 1 - \left( 1 - \frac{\varepsilon_c}{\varepsilon_{c2}} \right)^2 \right] \quad (4.15)$$

Aplicando a definição de E, como a razão entre a tensão e a deformação, tem-se o apresentado por (4.16):

$$E = \frac{d(\sigma_c)}{d\varepsilon_c} = \frac{d \left( 0,85 \cdot f_{cd} \cdot \left[ 1 - \left( 1 - \frac{\varepsilon_c}{\varepsilon_{c2}} \right)^2 \right] \right)}{d\varepsilon_c} \quad (4.16)$$

$$E = \frac{d(\sigma_c)}{d\varepsilon_c} = 0,85 \cdot f_{cd} \cdot \frac{2}{\varepsilon_{c2}^2} \cdot (\varepsilon_{c2} - \varepsilon_c) \quad (4.17)$$

Porém, para que a definição de módulo de elasticidade seja válida,  $\varepsilon_c \rightarrow 0$ , aplicando em (4.17), chega-se a relação (4.18). Apenas esclarecendo a notação:  $f_{cd,usado}$  é o  $f_{cd}$  a ser considerado pelo programa, para que este apresente módulo de elasticidade igual ao  $f_{ck,alvo}$ .

$$E(0) = 0,85 \cdot f_{cd,usado} \cdot \frac{2}{\varepsilon_{c2}^2} \cdot (\varepsilon_{c2}) = 0,85 \cdot 5600 \cdot \sqrt{f_{ck,alvo}}$$

$$f_{cd,usado} \cdot \frac{2}{\varepsilon_{c2}^2} = 5600 \cdot \sqrt{f_{ck,alvo}}$$

$$f_{cd,usado} = 5,6 \cdot \sqrt{f_{ck,alvo}} \quad (4.18)$$

Recordando,  $f_{cd}$  é o quociente entre  $f_{ck}$  e  $\gamma_c$ . Porém, é vantajoso manipular o valor de  $f_{cd}$  alterando o  $\gamma_c$ , pois o valor de  $f_{ck}$  também é usado para calcular a resistência à tração do concreto, usada para calcular a abertura de fissuras (ver seção 2.6.2 ou equação (2.22)). Logo,

mantendo o  $f_{ck}$  constante, o valor de  $\gamma_c$  que força os módulos de elasticidade definidos por (4.14) e (4.17) a serem iguais é dado por (4.20).

$$f_{cd,usado} = \frac{f_{ck,usado}}{\gamma_{c,usado}} = 5,6 \cdot \sqrt{f_{ck,alvo}} \quad (4.19)$$

$$\gamma_{c,usado} = \frac{\sqrt{f_{ck,alvo}}}{5,6} \quad (4.20)$$

Aplicando o  $f_{ck} = 20\text{MPa}$ , tem-se (4.21):

$$\gamma_{c,usado} = \frac{\sqrt{20}}{5,6} = 0,7986 \quad (4.21)$$

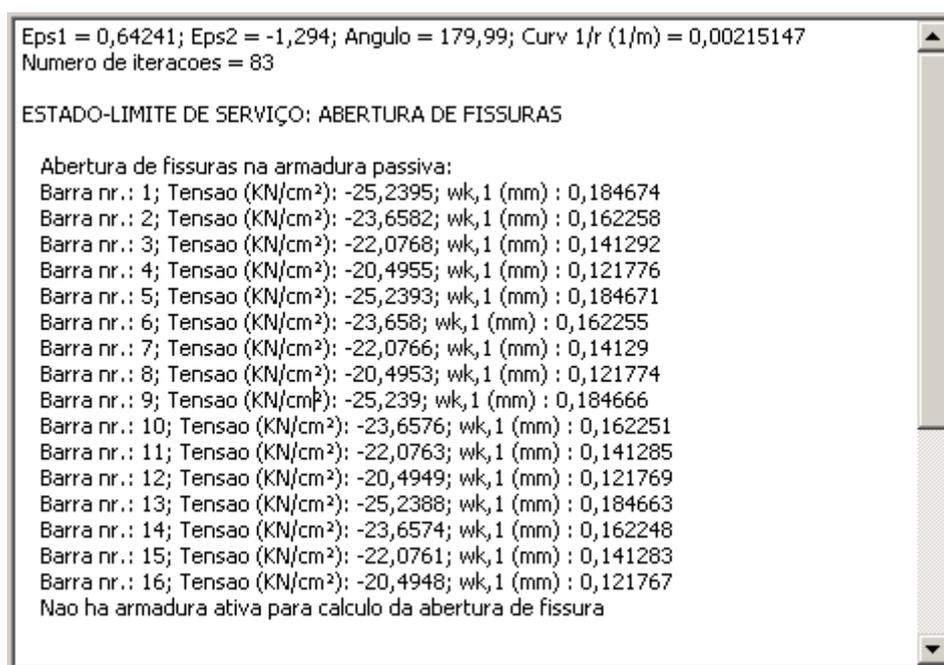
Finalmente, com os dados suficientes para considerar o módulo de elasticidade desejado pelas rotinas deste trabalho, aplicam-se os dados no programa:

- Pontos que definem a poligonal de concreto: ( 0 ; 0 ) , ( 25 ; 0 ) , ( 25 ; 90 ) , ( 0 ; 90 ).
- Pontos que definem as armaduras passivas: ( 4,28 ; 85,72 ) , ( 4,28 ; 82,22 ) , ( 4,28 ; 78,72 ) , ( 4,28 ; 75,22 ) , ( 8,58 ; 85,72 ) , ( 8,58 ; 82,22 ) , ( 8,58 ; 78,72 ) , ( 8,58 ; 75,22 ) , ( 16,42 ; 85,72 ) , ( 16,42 ; 82,22 ) , ( 16,42 ; 78,72 ) , ( 16,42 ; 75,22 ) , ( 20,72 ; 85,72 ) , ( 20,72 ; 82,22 ) , ( 20,72 ; 78,72 ) , ( 20,72 ; 75,22 ). A área de cada barra é de  $1,25\text{cm}^2$ .

Nos materiais:  $f_{ck} = 20\text{MPa}$  e  $\gamma_c = 0,7986$ . Os valores para armadura são os valores padrão para armadura passiva com aço CA-50 (ver seção 2.4)

O momento atuante na seção é de  $322,56\text{ kNm}$  (negativo), logo, no programa é inserido o valor de  $M_{xx} = -32256\text{ kN.cm}$ . Calculando, o programa retorna o seguinte resultado, mostrado pela Figura 63.

Figura 63 – Resultados da verificação do exemplo 12.



Os resultados segundo Carvalho e Figueiredo Filho (2007) são referentes à armadura na camada menos carregada (mais próxima da linha-neutra, cuja distância até a borda superior é 14,78cm – ver Figura 62). Segundo o autor, a tensão equivale a 204248 kN/m<sup>2</sup>, e a abertura de fissuras segundo a equação (2.22), equivale 0,12mm.

Como mostra a Figura 63 para as armaduras estudadas (que são as barras nro. 4, 8, 12 e 16), os resultados segundo a análise do presente trabalho são: para tensão, 204,955 kN/m<sup>2</sup>; e para abertura de fissura, 0,1218mm. Como mostrados, os resultados foram imensamente próximos (erro na ordem de 0,3%).

## 5 APRESENTAÇÃO DO USO DAS ROTINAS E DO PROGRAMA

No intuito de facilitar o reuso dos raciocínios e rotinas aqui apresentados, apresenta-se no presente capítulo recomendações e instruções para a importação destas.

Caso seja usada outra linguagem e/ou compilador, recomenda-se que as rotinas sejam reescritas, para que sejam respeitadas as particularidades de cada linguagem e evita-se assim dados corrompidos ou defeitos nos programas.

Se as rotinas aqui mostradas forem aplicadas na linguagem *Pascal*, com o compilador *Lazarus*, que foi o usado no presente trabalho e no qual as rotinas foram elaboradas, os códigos podem ser usados integralmente e sem alteração prévia: exceto caso deseje-se alterar um dos critérios e hipóteses apresentadas neste trabalho, mesmo assim, recomenda-se a leitura, compreensão, e possivelmente alteração das rotinas antes que estas sejam aplicadas, para que se conheça o comportamento da mesma.

Para incluir as rotinas citadas nesta seção em seu programa, basta adicionar ao projeto a *unit leoufscarfoc*. E na unidade principal, no ambiente *uses*, incluir a unidade *leoufscarfoc*.

### 5.1 ROTINAS;

As rotinas são separadas em uma ordem mais provável de uso, isto é, as rotinas preliminares de preparação dos dados para a iteração são mostradas primeiro, e as rotinas usadas ao finalizar o cálculo são mostradas ao final, no intuito de tornar a leitura mais intuitiva.

Outro ponto importante a se destacar é um tipo de variável usada neste programa: o tipo *TCoord*, para auxiliar as rotinas que usam coordenadas cartesianas. Tal tipo consiste em 2 números reais e um número inteiro, com as seguintes variáveis internas:

- *npol*: variável com função auxiliar, caso seja necessário enumerar os entes, sejam eles os pontos do polígono ou armadura

- *xi*: como o próprio título sugere, é o campo reservado para guardar um valor *x* do plano cartesiano;

- *yi*: analogamente ao anterior, campo reservado para guardar valor *y* no ponto cartesiano;

Destaca-se que podem ser pontos da poligonal que descreve a seção de concreto ou os pontos da armadura. Para a armadura, como são necessários mais dados, trabalha-se com outras variáveis em paralelo, para não criar vários tipos de variável.

### 5.1.1 Cálculo das características geométricas da seção;

Essa rotina foi condensada sob o nome padrão de *ObterCaractGeomet*, e é chamado e denominado no cabeçalho com o seguinte código: “*ObterCaractGeomet(const Coord:array of Tcoord; var fpAc,fpxcg, fpycg:real)*”.

Essa rotina aplica o descrito na seção 3.1.1, executando um procedimento que obtém a área e o centróide da figura descrita pelos pontos no vetor *Coord*, e guarda a área na variável *fpAc*, e as coordenadas *x* e *y* do centroide nas variáveis *fpxcg* e *fpycg*, respectivamente.

Ressalta-se que o vetor *Coord* deve apresentar os pontos de maneira que o último ponto e o primeiro deverão coincidir, pois ao executar o cálculo do *aux* do último ponto, será necessário o valor do primeiro (descrição na seção 3.1.1, equação do *aux* definida por (3.1)). Na descrição da rotina, pode-se verificar que ela é executada em todos os valores do vetor *Coord*, menos no último: exatamente por este ser um ponto complementar, isto é, construído apenas para coincidir com o primeiro.

### 5.1.2 Translação do eixo de coordenadas da seção;

A rotina de translação das coordenadas muda a posição do centro das coordenadas cartesianas que são usadas para definir a seção da barra, que é uma figura plana. Assim como mostra a seção 3.1.2, tal passo é importante para facilitar o cálculo dos momentos, principalmente, que serão considerados em relação ao centro geométrico da seção de concreto.

A rotina computacional é descrita no cabeçalho por “*procedure TranslacaoDePontos(const Coord:array of Tcoord; var fpxcg,fpycg:real;var CoordT:array of Tcoord)*” e executa o descrito nas equações (3.7) e (3.8).

As variáveis são: *Coord* é o vetor inicial, na posição inicial fornecida; *fpxcg* e *fpycg* são as coordenadas *x* e *y* que definem a posição do centro do novo eixo de coordenadas em relação ao primeiro, e coincidem com o centro geométrico da figura, por isso foi usada essa denominação; o *CoordT* é o vetor resultante da operação, que apresenta os pontos que definem o polígono em relação ao seu centro geométrico.

Deve-se definir o tamanho do vetor antes de executar essa rotina, verifica-se que o processo usado para isso no programa apresentado foi o *SetLength*, usado sempre que se aplica a rotina de translação, para que os vetores *Coord* e *CoordT* tenham o mesmo número de elementos.

### 5.1.3 Rotação do eixo de coordenadas da seção;

A rotina que executa a rotação dos pontos, descrita em 3.1.3, está indicada no arquivo pelo título “*procedure RotacaoDePontos(const Angulo:real;Coord:array of Tcoord; var CoordR:array of TCoord)*”, relacionada às expressões (3.9) e (3.10), que executa a rotação dos pontos em relação a origem no sentido anti-horário.

A cada ângulo analisado, a seção é girada, para que a linha-neutra seja sempre paralela ao eixo das abcissas, o que torna o cálculo das tensões mais simples, visto que suas expressões são em relação das deformações, e as deformações são estudadas como lineares a partir da linha-neutra. Santos (1994) apresenta o raciocínio, e embora a abordagem equacional difere da apresentada neste trabalho, é uma excelente fonte neste assunto. Já neste trabalho, a explicação para o uso da rotação é vinculada à seção 3.2, principalmente, visto que o cálculo das forças e momentos no concreto são muito facilitadas por essa rotação relativa da peça.

Ao executar a rotina, o usuário cita um ângulo, em graus, e inclui os pontos iniciais a serem girados em relação à origem  $(0,0)$ : tais pontos deverão estar no vetor *Coord*. O procedimento então insere as coordenadas em sua posição girada no vetor *CoordR*, que deverá ter o mesmo comprimento (número de elementos) que o vetor *Coord*, como citado ao final da seção imediatamente anterior a esta.

### 5.1.4 Cálculo das forças normais da seção de concreto;

A principal rotina deste trabalho, e também mais vulnerável e instável é a que calcula as forças e momentos no concreto, devido ao grande número de possibilidades, verificações quanto à posição do ponto em relação às possibilidades do comportamento tensão-deformação no concreto, e também devido à forma do polígono de concreto (que pode ser qualquer uma).

Para usá-la, a chamada é “*function ForcaConcreto(Poligono:array of Tcoord; ffcnfc, ffcnalfa, ffcnEpsilonC2,ffcnNconcreto:real):real*”. Neste procedimento, *Poligono* é o vetor que define o polígono de concreto, já no ângulo desejado, lembrando que a linha-neutra sempre passa horizontalmente na seção. A variável *ffcnfc* trata-se da resistência à compressão de cálculo do concreto  $f_{cd}$ . Já *ffcnalfa* deverá calcular a distância da linha neutra e do maior ponto vertical da seção  $x_a$ , obtido na equação (3.17) e mostrado na seção 3.2. Paralelamente, a variável *ffcnEpsilonC2* deve incluir o comprimento  $x_{ec2}$ , que corresponde ao comprimento em que as deformações estão entre zero e  $\varepsilon_{c2}$  (mais informações na seção 3.2), comprimento este calculado na equação (3.18) e mostrado na seção 3.2). E por último, a variável *ffcnNconcreto* deverá apresentar a variável  $n$ , que diz respeito à relação tensão-

deformação do concreto, anexa à Figura 9. As unidades usadas não precisam estar num arranjo pré-fixado, apenas condizentes umas com as outras.

Tal rotina corresponde a uma função e não a um procedimento, o que possibilita manejo mais fácil do programador, cuja abordagem foi julgada mais adequada para esta operação. É um alerta que se faz é que a função só opera se a borda superior for mais comprimida que a inferior ( $\varepsilon_1 > \varepsilon_2$ ), e caso seja necessário fazer a borda superior mais tracionada, deve-se girar a seção 180° e tornar a borda inferior (que antes era a superior) mais tracionada.

### 5.1.5 Cálculo dos momentos da seção de concreto;

As funções desta seção calculam os momentos resistentes por uma seção poligonal de concreto, funcionando de maneira quase idêntica ao cálculo das forças normais: a diferença está apenas na resposta fornecida pela função apenas, pois como o título indica, a função desta seção fornece um resultado na solicitação momento, que é calculado em função da origem do sistema de coordenadas. Por isso, como se cita nas seções 3.1.2 e 5.1.2, a origem do sistema de coordenadas e o centro geométrico da seção coincidem.

A chamada da função que retorna o momento em relação a  $x$  é “*function MomentoXXConcreto (Poligono:array of Tcoord; ffcnfc, ffcxalfa, ffcxepsilonC2, ffcnNconcreto:real):real;*”.

E a chamada da função que retorna o momento em relação a  $y$  é “*function MomentoYYConcreto (Poligono: array of Tcoord; ffcnfc, ffcxalfa, ffcxepsilonC2, ffcnNconcreto:real):real;*” da seção.

Ambas as funções têm variáveis com a mesma nomenclatura, que também é igual àquela usada na função que calcula a força na seção de concreto, na seção 5.1.3, portanto sugere-se a consulta deste para utilização correta das funções de momento apresentadas.

### 5.1.6 Cálculo da tensão na armadura passiva;

A função aqui apresentada retorna para o usuário a tensão na armadura passiva e segue rigorosamente o gráfico mostrado na Figura 10, isto é, uma função linear entre os limites  $-\varepsilon_{yd}$  e  $\varepsilon_{yd}$  e fora desses limites tem o valor de  $-f_{yd}$ , quando a deformação é menor que  $-\varepsilon_{yd}$  e  $+f_{yd}$  quando a deformação é maior que  $\varepsilon_{yd}$ .

Sua chamada é feita pelas palavras chave “*function TensaoArmPas( var varEpsilonS, varEs, varfyd: Real):real*”, em que *varEpsilonS* é o valor de  $\varepsilon_s$ , isto é, o valor ao qual a armadura está sujeito; *varEs* é o valor do módulo de elasticidade na região não constante do gráfico e *varfyd* é o valor do  $f_{yd}$ , isto é, da resistência ao escoamento de cálculo da armadura

passiva. As unidades devem estar compatíveis umas com as outras, a única exceção é a deformação, que deve ser usada em 1/1000 (simplificando: uma deformação de 0,003 deverá ser 3).

Outro fator importante é que não se julgou necessária uma função ou um procedimento para calcular diretamente a força ou momento na armadura passiva pois é bastante simples fazê-lo a partir da tensão, sendo que a elaboração de tal rotina não proveria uma facilitação efetiva nas operações, que consistem em pouquíssimas linhas de programação, que são apresentados na seção 3.4 e nas linhas do programa principal.

### 5.1.7 Cálculo da tensão na armadura ativa;

A função da tensão na armadura ativa é chamada pela indicação: “*function TensaoArmAtiv (var varEpsilonpyd, varEpsilonpu, varfpyd, varfptd, varEpsilont: Real): real*”. Tal função localiza a tensão análoga a uma certa deformação na relação tensão-deformação da armadura ativa. A deformação aplicada na função (deformação total) deve considerar o pré-alongamento da armadura ativa e a deformação na seção de concreto, como é apresentado na equação (2.12).

Tratando especificamente das variáveis, tem-se: *varEpsilonpyd* é o valor de deformação de início do escoamento de cálculo  $\epsilon_{pyd}$ , *varEpsilonpu* o valor de deformação do aço na ruptura, *varfpyd* é o valor da tensão da armadura no início do escoamento, *varfptd* é o valor de tensão de ruptura na armadura e *varEpsilont* é o valor de deformação total (soma do pré-alongamento e deformação atuante na seção). As unidades usadas devem ser compatíveis entre si, e recomenda-se usar a deformação em termos de 1/1000, assim como na armadura passiva.

Assim como no caso de armadura passiva, não foram elaboradas funções para obtenção direta da força e momento, visto que são facilmente obtidas a partir de sua deformação, e que pouco facilitariam eventuais reutilização dessas rotinas.

## 5.2 RESUMO DAS FUNÇÕES APRESENTADAS E SEUS PARÂMETROS E VARIÁVEIS;

Apresentam-se o resumo das funções na tabela, e para maiores informações, recomenda-se a leitura do tópico relativo à função citada. As funções apresentadas na tabela 1 são usadas diretamente para o cálculo do estado-limite último. Além destas, há também rotinas que são de uso mais difícil ou que foram usadas como funções auxiliares, que compõem outras funções usadas, que estão na tabela 2.

Tabela 7. Rotinas prontas com detalhamento.

Função/Procedimento	Descrição
ObterCaractGeomet	Calcula área, e coordenadas $x$ e $y$ do centro de gravidade da seção. Internamente, a rotina calcula o momento estático nas direções $x$ e $y$ também.
TranslacaoDePontos	Faz a translação das coordenadas, fazendo com que o novo centro seja nas coordenadas impostas (que podem ou não ser do centro de gravidade)
RotacaoDePontos	Essa rotina faz uma rotação da figura poligonal no sentido anti-horário, isto é, gira o eixo de coordenadas no sentido horário em relação à figura. O ângulo deve estar em graus.
ForcaConcreto	Rotina usada para calcular a força no concreto segundo certa conformação de deformações. A entrada de dados deve incluir o comprimento de certas etapas de tensões no concreto: por exemplo, a altura da linha neutra, ou o comprimento
MomentoXXConcreto	A rotina usa os dados de maneira semelhante à rotina ForcaConcreto, e calcula o momento tendo como referência o eixo $x$
MomentoYYConcreto	A rotina usa os dados de maneira semelhante às rotinas ForcaConcreto e MomentoXXConcreto, e calcula o momento tendo como referência o eixo $y$
TensaoArmPas	Faz-se necessário também o cálculo das forças e momentos do aço, e para isto não foram geradas várias funções, mas apenas a de tensão, como função das propriedades do material e da deformação
TensaoArmAtiv	Além disso, calcula-se as tensões devido à armadura ativa, como descreve a seção 3.4, em função da deformação na armadura e de suas características mecânicas

Além destas funções e procedimentos, apresenta-se também a tabela 2, que apresenta rotinas de uso mais geral, ou que fizeram parte da programação de outras rotinas. Sugere-se que caso forem aplicadas, sejam avaliadas de antemão, visto que fazem totalmente parte do esqueleto de outras rotinas.

Tabela 8. Funções e procedimentos secundários.

Função/Procedimento	Descrição
Ncd1	Resolve a equação (3.21), parte integrante das funções ForcaConcreto, MomentoXXConcreto e MomentoYYConcreto.
Mcd1xx	De maneira semelhante à Ncd1, porém referente à equação (3.22).
Mcd1yy	De maneira semelhante à Ncd1, porém referente à equação (3.23)
Ncd2	De maneira semelhante à Ncd1, porém referente à equação (3.24)
Mcd2xx	De maneira semelhante à Ncd1, porém referente à equação (3.25)
Mcd2yy	De maneira semelhante à Ncd1, porém referente à equação (3.26)
PontosMaximoMinimo	Retorna o maior e o menor resultado no campo $y_i$ de um vetor de coordenadas do tipo TCoord, que foi o tipo usado no programa

### 5.3 PROCEDIMENTOS PARA EXPORTAR RESULTADOS DO PROGRAMA

Outro aspecto deste trabalho de menção importante são as utilidades do programa. Embora as rotinas possibilitem um reaproveitamento mais fácil do que foi apresentado, entender o programa permite que as rotinas sejam testadas e que se utilize o programa.

Devido às dificuldades ligadas à programação, a exportação dos dados ainda não pode ser feita de maneira ideal, por exemplo: exportar os diagramas inteiros de ELU em programas de gráficos e desenhos; exportar os dados gerados no gráfico para planilhas confeccionadas automaticamente. Isso facilitaria muito ao usuário. Ou ainda, num cenário ainda mais abrangente, a possibilidade de exportar as seções e/ou seus resultados para outros programas de análise de estruturas, tornando fácil e rápida a comparação entre diferentes programas.

A exportação dos dados é realizada com auxílio de um campo de texto simples. Este campo usa sempre os mesmos caracteres, como “=” (sinal de igual) e “;” (ponto e vírgula), e apresenta o resultado de cada ponto em uma linha isolada. Para demonstrar como manusear estes dados, toma-se o caso do exemplo 7: Neste exemplo, a curvatura é convertida em curvatura adimensional, como propõe a equação (4.5), fazendo uso de uma planilha, que torna o cálculo automático. Na Figura 64 é possível identificar etapas no processo de manuseio dos dados:

Com os dados no programa (material, seção de concreto, armadura passiva e força normal), ativa-se o botão “Calcular” o programa gera o gráfico (se o cálculo for propício, como para ELU e Momento-Curvatura) e campo de texto. Obtém-se assim os dados no programa, descritos pelo primeiro campo da Figura 64.

Então, pode-se exportar os dados para outro campo de texto ou planilha, usando o próprio programa, ou ainda selecionando-os e copiando-os. Os resultados em um outro ambiente se apresenta no segundo campo da Figura 64. O próprio programa deve ser capaz de salvar o texto automaticamente num arquivo, para que facilite o manuseio do programa e seus resultados.

Figura 64 – Resultados do programa sendo copiados para uma planilha.

Eps1=2,57348; Eps2=-9,72652; força=0,0000135446; Curv.=0,0492; Mxx = 7463,39; Myy = 0
Eps1=2,58753; Eps2=-9,81247; força=3,54767E-6; Curv.=0,0496; Mxx = 7463,89; Myy = 0
Eps1=2,60157; Eps2=-9,89843; força=1,26278E-6; Curv.=0,05; Mxx = 7464,37; Myy = 0
Eps1=2,6156; Eps2=-9,9844; força=0,0000137864; Curv.=0,0504; Mxx = 7464,85; Myy = 0

Porém, os dados ainda estão apenas como linhas de texto, e não no formato de planilha, onde cada célula possui um dado isolado. Para alterar o formato, usa-se uma ferramenta de transformação de texto em colunas, especificando como caracteres que dividem as colunas “;” e “=”, e assim obtém-se o apresentado inicialmente na Figura 65.

Com as colunas divididas, basta eliminar as que têm conteúdo indesejado. Por exemplo, na mesma figura, foram eliminadas as colunas com os nomes das variáveis e os valores da variável  $M_{yy}$ , que foi sempre *zero*, pois a seção é simétrica. Então os dados estão prontos para serem utilizados de qualquer forma que o usuário desejar, por exemplo, como apresenta a mesma figura, faz-se a conversão da curvatura  $1/r$  em curvatura adimensional  $\theta$ , usando a equação (4.5), como mostra a imagem inferior da Figura 65:

Figura 65 – Resultados do programa aplicados e modificados numa planilha.

Eps1	2,57348	Eps2	-9,72652	força	-1,4E-05	Curv.	0,0492	Mxx	7463,39	Myy	0
Eps1	2,58753	Eps2	-9,81247	força	3,55E-06	Curv.	0,0496	Mxx	7463,89	Myy	0
Eps1	2,60157	Eps2	-9,89843	força	1,26E-06	Curv.	0,05	Mxx	7464,37	Myy	0
Eps1	2,6156	Eps2	-9,9844	força	1,38E-05	Curv.	0,0504	Mxx	7464,85	Myy	0

$\epsilon_1$	$\epsilon_2$	N	$1/r$	Mxx	$\theta$	M
2,58753	-9,81247	3,55E-06	0,0496	7463,89	12,4	74,64
2,60157	-9,89843	1,26E-06	0,05	7464,37	12,5	74,64
2,6156	-9,9844	1,38E-05	0,0504	7464,85	12,6	74,65

Embora a exportação dos resultados já é possível, há a sugestão de melhorá-la. Na seção 6.1 que trata especificamente de estudos futuros, há sugestões para melhoria do programa em si também, incluindo sugestões específicas sobre exportação e análise dos resultados obtidos no programa, além de outras maneiras de exportação de dados já citados nesta mesma seção.

## 6 CONSIDERAÇÕES FINAIS

O trabalho apresenta maneiras de se analisar seções sujeitas à força axial e momento fletor em qualquer direção na seção, que caracteriza a flexão oblíqua composta. Segundo a bibliografia e como se apresenta neste trabalho, o cálculo iterativo possibilita a avaliação de seções de concreto sujeitas à flexão oblíqua composta tanto em estado-limite último quando em situações de serviço, incluindo análises com armadura ativa.

Trata-se de um programa numérico analisa a seção a partir de uma seção de concreto definida, isto é, estabelecida a geometria da seção de concreto (aproximada a um polígono) e da armadura passiva e/ou ativa, além das características desses materiais, o programa pode realizar as diferentes análises: a análise de estado-limite último, em que obtém-se a envoltória de momentos fletores resistentes da seção; a análise momento-curvatura, em que obtém-se a relação momento-curvatura para a seção diante de uma força normal; e a análise de estado-limite de serviço, que analisa as deformações na seção diante de esforços normais estabelecidos, e compara os resultados às prescrições normativas.

Para avaliar a seção quanto ao estado-limite último, resumidamente, calcula-se uma envoltória de momentos resistentes apresentados pela seção. Para isso, varia-se a profundidade da linha neutra na seção, procurando a profundidade que corresponde à força normal atuante. Quando for igualada, o momento apresentado pela seção é o momento resistido em tal direção, que deverá ser registrado para desenhar o diagrama. A força normal poderá encontrar-se entre 2 extremos: força normal máxima de compressão na seção, e a máxima de tração na seção, e entre eles a seção oferecerá resistência a momentos atuantes também. As operações adotadas foram seguidas conforme os passos sugeridos por Santos (1996) e Fusco (1981), que apresentam a análise de ELU para FOC de maneira muito didática.

A avaliação da relação momento-curvatura é adaptada da realizada em estado-limite último: fixa-se a diferença de deformação entre dois pontos extremos da seção (fixando assim, a curvatura). Calcula-se a força normal na seção respeitando o valor da curvatura, procurando a força normal desejada num processo numérico iterativo. Ao encontrar a conformação de deformações correspondente à força normal desejada, calcula-se o momento apresentado pela seção: obtém-se assim um ponto no diagrama momento-curvatura, pra determinada força axial. Ao variar o valor da curvatura e efetuar o cálculo novamente, obtém-se outro ponto do diagrama, e sucessivamente, até que os valores de deformações esgotem a capacidade dos

materiais, e calcula-se o diagrama completo pra determinado ângulo. Fusco (1981) apresenta um algoritmo com orientação semelhante.

Já a avaliação de serviço é de elaboração própria, inspirada nos conceitos apresentados por Fusco (1981) para curvatura em qualquer direção, decomposta em duas partes: curvatura em  $x$  e curvatura em  $y$  e nos conceitos apresentados por Santos (1996) para cálculo das solicitações nas seções. Aplicam-se métodos numéricos para convergir para a solução: isto é, descobrir quais as deformações que resultam em determinadas solicitações. Com as deformações, pode-se calcular as tensões em qualquer posição da seção (incluindo suas armaduras), e portanto, podendo calcular os estados-limites de serviço ELS-W, ELS-F e ELS-D. O único problema é a avaliação da fissuração (ELS-F), pois a avaliação da norma e a abordagem do trabalho não são perfeitamente compatíveis – a norma utiliza a equação do momento de fissuração, e o trabalho apenas analisa a deformação máxima de tração que ocorre na seção, e compara com o valor limite de tração do concreto. Maiores informação são apresentados na seção 2.6.3.

Quanto à análise numérica e a convergência do programa para o resultado, usado para resolução das rotinas de ELU e obtenção da relação momento-curvatura, se sugere aplicar o método das secantes. Esse método numérico atende a necessidade, isto é, apresenta convergência com um grau de convergência médio, pois é maior que linear, porém menor que quadrático.

Já para resolução do sistema de equação – caso da rotina dos estados-limites de serviço – sugere-se aplicar o método de Newton com duas adaptações: devido à dificuldade em diferenciar a função (muitas variáveis e condições diferentes), sugere-se usar uma discretização – isto é, o Método de Newton discretizado, que substitui o cálculo das derivadas de maneira semelhante ao Método das Secantes. Além disso, para evitar comportamentos inesperados e a dificuldade de se trabalhar com matrizes a cada iteração, limita-se a matriz Jacobiana a um valor constante, calculando seu valor apenas uma vez. Embora o grau de convergência seja baixo, o método chegou ao resultado desta análise mais complexa, que é partir das solicitações e obter as deformações (sistema de equações não linear).

Salienta-se a divergência entre os enfoques das bibliografias no uso ou não de variáveis adimensionais (como os coeficientes  $\nu$ ,  $\mu_x$  e  $\mu_y$ ), mostrados nos exemplos de validação 2 e 3, seção 4.2 e 4.3. E sugere-se enfoque principal no não uso dessas variáveis adimensionais, pois com os recursos computacionais atuais é possível executar o cálculo sem a necessidade destas e com liberdade de conformações de seção, pois os ábacos exigem proporções fixas.

Outro fator extremamente relevante no desenvolvimento deste trabalho é a equação da norma para a relação tensão-deformação no concreto (mostrada na Figura 9). A relação tensão-deformação explicitada como função possibilita a integração que é fundamental pros métodos aplicados. A norma ABNT NBR 6118:2014 apresenta uma diferença nesta equação em relação à sua antecessora: permite considerar a resistência do concreto à compressão superior à 50 MPa.

Para isso, o expoente que na equação da relação tensão deformação do concreto (seção 2.4) anterior 2 deve ser substituído por um coeficiente  $n$  caso a resistência característica a compressão do concreto for maior que 50 MPa. Este expoente aumentou bastante a dimensão das derivadas das equações, adicionando várias linhas à estas: isso é indesejado, pois além de torná-las mais trabalhosas, pode também tornar sua integração exata impossível, forçando o uso de métodos numéricos, que acarretam presença de erro e/ou maior trabalho computacional. Assim, reduzir a equação que descreve a relação tensão-deformação do concreto é oportuno nesta área de estudo.

Concluindo, as rotinas seguem o escopo deste trabalho, portanto, tem foco na análise de seções submetidas à flexão composta oblíqua. Porém, essas rotinas podem ser utilizadas em outras situações. Por exemplo, uma rotina de verificação de estado-limite último pode permitir a elaboração de uma rotina de dimensionamento – propondo uma seção com armadura, e aumentando a armadura cada vez que os momentos resistentes não atenderem os momentos de cálculo. A rotina para obtenção da relação momento-normal-curvatura pode ser utilizada para considerar efeitos de não linearidade - e a rotina pode gerar o gráfico para  $0,85f_{cd}$  ou  $1,1f_{cd}$ , como proposto pela seção 15.2.1 da ABNT NBR 6118:2014, e mostrado no exemplo 6.

As rotinas de estado-limite de serviço podem ser aplicadas para se calcular a deformação análoga à certos esforços resistentes – isso permite que se calcule certos estados-limite de serviço, ou ainda como uma parte do método geral (para consideração da relação momento-curvatura real, como explica a norma ABNT NBR 6118:2014 na seção 15.8.3.2).

Além da contribuição das rotinas, há também o programa OblqCALCO, que calcula as envoltórias de momentos resistentes em relação a uma força axial constante, em estado-limite último; calcula as relações momento-normal-curvatura da seção; calcula as deformações oriundas de certos esforços solicitantes, calculando os estados-limite de abertura de fissuras, formação de fissuras e descompressão.

## 6.1 SUGESTÕES PARA TRABALHOS FUTUROS

A partir do apresentado, observam-se pontos no programa e nas rotinas cujo desenvolvimento traria grande benefício ao usuário, com foco inicial em medidas defensivas (rotinas que não permitam dados incorretos a serem inseridos ou que corrijam eventuais erros), outro campo bastante relevante para o trabalho com flexão oblíqua composta no geral seria melhorar as formulas em si. Mais uma utilidade é criar rotinas que vão além destas criadas: por exemplo, que desenhe em 3D o diagrama de estado-limite último.

Todas essas sugestões estão discriminadas a seguir, com sugestões para elaboração e detalhes úteis:

- Medida defensiva: A utilização da seção descrita por uma figura poligonal facilita muito os cálculos, mas também leva a outras dificuldades: a primeira a ser destacada é que os pontos sequencialmente em sentido anti-horário descrevem uma área positiva. Já os pontos em disposição horária simbolizam área negativa. Dessa forma, para desenhar uma seção vazada, deve-se desenhar a seção externa usando uma sequência de pontos em sentido anti-horário e então desenhar a parte vazia com uma poligonal interna no sentido horário, como apresenta a Figura 19.

Porém, numa situação de seção normal, se o usuário inserir os dados na ordem horária, é conveniente que o programa faça a inversão automaticamente. Para reconhecer, basta calcular a área da poligonal: caso seja negativa, o programa deve inverter a poligonal automaticamente.

Mais uma sugestão que se faz em relação quantos às medidas defensivas referentes à entrada de dados em si é sobre a posição das armaduras: não foram citadas maneiras de avaliar se as armaduras estão dentro da área útil da poligonal de concreto. A dificuldade principal é no caso da armadura estar no interior de uma seção vazada. Inclusive, uma rotina que avalie a posição das armaduras e citasse ao usuário o cobrimento efetivo na peça seria ainda melhor, resolvendo o problema citado nesse parágrafo e fornecendo mais um dado útil nas avaliações.

- Melhoria na entrada de dados: Os dados que descrevem a seção são informados ao programa usando a poligonal. Porém, muitas vezes a seção utilizada segue um padrão frequentemente usado: como seções retangulares, seção T, seção I, entre outras. Sugere-se que seja incluída no programa uma entrada rápida pra essas seções padrão, inclusive podendo ou não adicionar armadura, sendo que o usuário deve apresentar o cobrimento, naturalmente.

Essa rotina pode ser elaborada em etapas, sendo um padrão de seção incluído de cada vez. Pode-se até mesmo elaborar uma entrada rápida apenas para a seção de concreto a princípio, e posteriormente incluir armaduras.

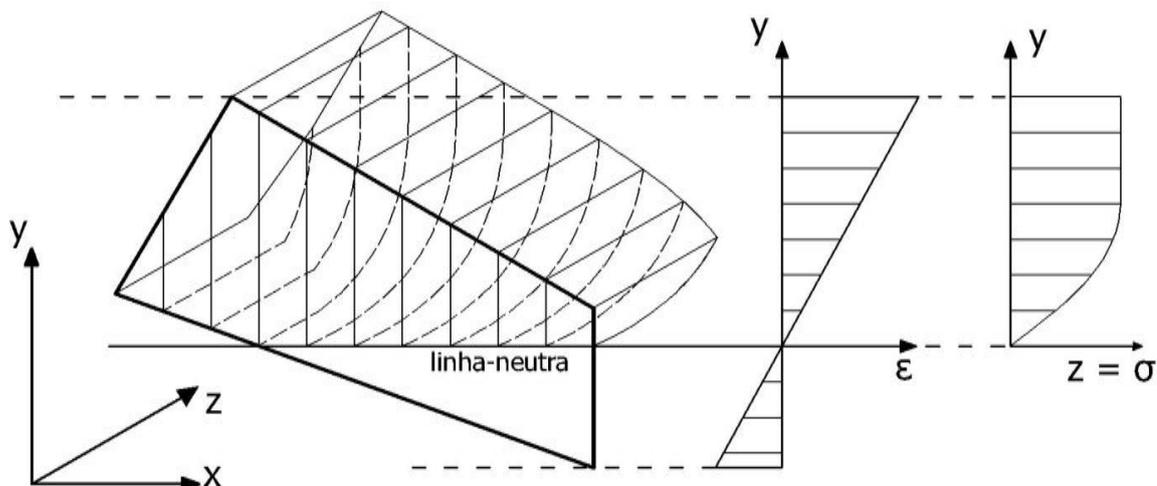
- Melhoria das equações: O processo das equações tratadas é descrito especificamente na seção 3.2. São as equações referentes à poligonal e à força do concreto. Uma poligonal é dividida em várias retas. Essas retas são registradas como equações  $x = f(y) = a y + b$ . Deve se levar em conta que a função padrão  $y = f(x)$  foi trocada em sua nomenclatura para  $x = f(y)$  para permitir os cálculos propostos. Qualquer dúvida sobre este assunto especificamente, recomenda-se também a leitura da seção 3.2, e Santos (1994) para mais informações.

Numa seção retangular, sabe-se que há trechos em que  $x_n \neq x_{n+1}$  e  $y_n = y_{n+1}$ . Essa reta, seguindo o equacionamento citado, apresenta  $a = \pm \infty$ , pois o denominador se aproxima de zero. Se o denominador for zero, o programa trava, e retorna um erro. Para evitar isso, o programa faz com que se evite as regiões em que  $y_n = y_{n+1}$ , pois a força de concreto seria zero de qualquer maneira ( $\text{área} = 0$ , pois  $dy = 0$ , em relação ao padrão de cálculo mostrado, ver Figura 21). É possível escrever essas retas como sendo pontos  $(x_p, y_p)$  + vetores  $(x_v, y_v)$ . De maneira que o módulo do vetor seja igual ao comprimento da reta no polígono. É provável que tal mudança evite que os cálculos apresentem o ponto fraco citado.

- Melhoria nas equações: Outra perspectiva para se considerar a força de concreto na área descrita pelo polígono de concreto – melhor descrição na sugestão anterior, na seção 3.2 e na bibliografia Santos (1994) – é que se calcule o produto entre tensão e área não mais como foi apresentado, mas como uma integral tripla. Considerando a força como somatório do produto de partículas infinitesimais de área por sua respectiva tensão, pode-se ter a perspectiva de um plano tridimensional com as coordenadas  $x$ ,  $y$  e  $z$  sendo que  $x$  e  $y$  definem o polígono de concreto e  $z$  define a tensão numa posição específica. A equação que representa esse conceito é apresentada a seguir por (6.1) e pela Figura 66.

$$F = \int_A \sigma \cdot dA = \iiint_V dx dy dz \quad (6.1)$$

Figura 66 – Cálculo da força no concreto por integração tripla.



- Exportação: O estado-limite último é apresentado em forma gráfica no ambiente de ELU. No geral, o diagrama é apresentado segundo um carregamento axial constante, formando um gráfico em 2D (duas dimensões). Porém, para apresentar um diagrama único que represente todas as possibilidades de carga em ELU, é preciso unir todos os diagramas, formando uma figura tridimensional (3D) como mostra a Figura 12.

Com tais rotinas, é possível obter-se cada ponto desejado do diagrama, logo, com o recurso computacional correto, é possível também desenhar o diagrama em 3D: essa é a sugestão, a elaboração de uma rotina que exponha os pontos num diagrama tridimensional. Inclusive, o programa além de gerar o gráfico com cada ponto, extrai os resultados dos momentos calculados num campo de texto: esse campo de texto é um exemplo de como exportar os dados gerados pelo programa. Também se pode extrair num arquivo externo, com uma extensão qualquer. Algumas extensões conhecidas para transferir dados tridimensionais são *dxg*, *csv* e *xyz*, porém não faz parte do escopo deste trabalho a maneira de se desenhar e compor tais arquivos.

## 7 REFERÊNCIAS

ARAÚJO, J. M. **Dimensionamento à flexão simples de seções transversais de concreto armado com ênfase na ductilidade. Teoria e prática na engenharia civil.** n.14. p. 1-13, out 2009. Disponível em <[http://www.editoradunas.com.br/revistatpec/Art1\\_N14.pdf](http://www.editoradunas.com.br/revistatpec/Art1_N14.pdf)>. Acesso em nov, 2013.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6118:2014.** Projeto de estruturas de concreto: Rio de Janeiro 2014.

CARDOSO JÚNIOR, S. D. ; KIMURA, A. E. . Sistema computacional para análise não linear de pilares de concreto armado. In: CONGRESSO BRASILEIRO DO CONCRETO, 55., 2013, Gramado. **Anais do 55º Congresso Brasileiro do Congresso**, 2013.

CARVALHO, R. C; FIGUEIREDO FILHO, J. R. **Cálculo e Detalhamento de Estruturas Usuais de Concreto Armado.** Segundo a NBR 6118:2003. 3 ed. Editora da Universidade Federal de São Carlos EdUFSCar. 2007.

CARVALHO, R. C. **Estruturas de Concreto Protendido:** cálculo e detalhamento. 1 ed. Editora Pini. São Paulo, SP. 2012.

CECCON, J. L. **Análise dos efeitos de segunda ordem em pilares solicitados a flexão oblíqua composta.** 306f. Tese (Doutorado em Estruturas) – Escola Politécnica, Universidade de São Paulo. São Paulo, 2008.

Concreto Armado. Versão 4.0. São Carlos: UFSCar. 2009.

FALEIROS JUNIOR, J. H. **Procedimento de Cálculo, verificação e detalhamento de armaduras longitudinais na seção transversal em elementos protendidos.** 179 f. Dissertação (Mestrado em Construção Civil) – Universidade Federal de São Carlos. São Carlos, 2010.

FRANCO, Neide Bertoldi. **Cálculo Numérico.** São Paulo: Prentice Hall, 2006.

FTOOL – Two-Dimensional Frame Analysis Tool. Versão Educacional 3.0. Rio de Janeiro: Tecgraf/PUC-RIO, 2012. Disponível em <[www.tecgraf.puc-rio.br/ftool](http://www.tecgraf.puc-rio.br/ftool)>. Acesso em ago 2013.

FUSCO, Péricles Brasiliense. **Estruturas de concreto – solicitações normais.** Rio de Janeiro: Editora Guanabara Dois S.A., 1981.

GREGORI J. N.; SOSA P. M.; FERNÁNDEZ-PRADA, M.A; FILIPPOU , F. C. A 3D numerical model for reinforced and prestressed concrete elements subjected to combined axial, bending, shear and torsion loading. **Engineering Structures.** v. 29, p. 3404-3419, dez. 2007. Disponível em <[dx.doi.org/10.1016/j.engstruct.2007.09.001](http://dx.doi.org/10.1016/j.engstruct.2007.09.001)>. Acesso em ago. 2013.

LANGENDONCK, Telemaco Van. **Flexão composta oblíqua no concreto armado.** São Paulo: EESC – USP, 1977.

LEONHARDT, F; MÖNNIG, E. **Construções de Concreto: Princípios básicos do dimensionamento de estruturas de concreto armado**. Vol 1. Editora Interciência Ltda. Rio de Janeiro , RJ. 1977.

LEONHARDT, F; MÖNNIG, E. **Construções de Concreto: Verificação da capacidade de utilização, limitação da fissuração, deformações, redistribuição de momentos e teoria das linhas de ruptura e estruturas de concreto armado**. Vol 4. Editora Interciência Ltda. Rio de Janeiro , RJ. 1979.

MARINO, M. A. **Seções transversais de concreto armado sujeitas a solicitações normais**. Curitiba: COPEL, 1979.

MEDEIROS, G.A. **Programa para análise e dimensionamento da área de armadura das seções de concreto armado submetidas a flexão oblíqua composta**. Trabalho de Graduação – Divisa de Engenharia de Infra-estrutura Aeronáutica. 2004. São José dos Campos, SP.

MENDES NETO, F. **Concreto estrutural avançado - Análise de Seções Transversais sob Flexão Normal Composta**. 1.ed., Editora PINI, 2010.

MONTOYA, P. J; MESEGUER, A. G.; CABRÉ, F. M. **Hormigón armado**. 13ª Edição. Editora Gustavo Gili. Barcelona, 1991.

NORMAL. Versão 1.3. Curitiba: UFPR, 2001.

OBLÍQUA. Versão 1.0. Curitiba: UFPR, 2001.

PALLARÉS, L.; MIGUEL , P. F.; FERNÁNDEZ-PRADA, M. A. A numerical method to design reinforced concrete sections subjected to axial forces and biaxial bending based on ultimate strain limits. **Engineering Structures**, v. 31, p. 3065-3071, dez. 2009. Disponível em <[dx.doi.org/10.1016/j.engstruct.2009.08.006](https://doi.org/10.1016/j.engstruct.2009.08.006)> . Acesso em ago. 2013.

PAULA, J. A. **Algoritmos para o estudo de pilares esbeltos de concreto solicitados à flexão normal composta**. 201p. Dissertação (Mestrado em Engenharia de Estruturas). Escola de Engenharia de São Carlos, Universidade de São Paulo. São Carlos, SP. 1988.

PCalc. Versão 1.1. Sander David Cardoso. 2013.

PINHEIRO, L. M; BARALDI, L. T; POREM, M. E. **Estruturas de concreto: ábacos para flexão oblíqua**. Departamento de Engenharia de Estruturas. Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2009.

RIBEIRO, K. **Diagramas para verificação de pilares retangulares em concreto armado submetidos à flexão composta normal**. 308p. Dissertação (Mestrado em Engenharia Civil) – Programa de Pós-Graduação em Engenharia Civil. Universidade Federal de Santa Catarina. Florianópolis, 2011.

RUGGIERO, M. A. G; LOPES, V. L. **Cálculo numérico: aspectos teóricos e computacionais**. 2ª Ed. Pearson Makron Books. São Paulo, SP. 2006.

SMANIOTTO, Alberto. **Dimensionamento e detalhamento automático de pilares retangulares submetidos à flexão composta oblíqua.** 120p. Dissertação (Mestrado em Engenharia Civil) – Programa de Pós-Graduação em Engenharia Civil. Universidade Federal de Santa Catarina. Florianópolis, 2005.

SANTOS, L. M. **Sub-rotinas básicas do dimensionamento de concreto armado.** v.1. São Paulo: Thot Editora, 1994.

SANTOS, S. H. C. **CONSIDERAÇÕES SOBRE ASPECTOS NORMATIVOS DO PROJETO DE ESTRUTURAS DE CONCRETO.** Palestra ministradas no 55º Ibracon, Gramado, 2013.

VENTURINI, W. S; RODRIGUES, R. O. **Dimensionamento de peças retangulares de concreto armado solicitados à flexão reta.** Departamento de Engenharia de Estruturas. Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 1987.

Wolfram Alpha LLC. **Wolfram|Alpha.** 2014.

## **APÊNDICES**

## A. APÊNDICE 1 – MANUAL DO PROGRAMA OBLQCALCO

### A.1 CARACTERÍSTICAS GERAIS E ENTRADA DE DADOS

Este programa tem como foco resolver problemas envolvendo esforços normais, portanto aqui se desprezam as solicitações tangenciais. As solicitações avaliadas são forças axiais e momento fletor em qualquer ângulo.

O programa divide-se em 4 abas, isto é, 4 iterações com diferentes objetivos:

- ELU: Gera um gráfico que mostra as solicitações que a seção suporta com segurança, apresentando a altura da linha neutra de ruptura também;
- Momento-Curvatura: Gera um gráfico que relaciona a grandeza momento com a grandeza curvatura, sob uma força axial resultante fixa na seção;
- Calcular Solicitações: Nessa aba, o programa calcula a partir dos dados fornecidos pelo usuário: ângulo, deformação na fibra superior e deformação na fibra inferior, calcula as solicitações correspondentes;
- Verificar Solicitações / ELS: A partir de solicitações escolhidas pelo usuário, calcula o estado de deformações correspondente (ângulo e deformações nas bordas inferior e superior) e verifica os 3 estados-limite de serviço: abertura de fissuras (ELS-W), formação de fissuras (ELS-F, parcialmente alterado, sugere-se consulta) e descompressão (ELS-D).

A seguir, se apresenta a imagem inicial do programa na Figura A. 1, que é o principal ambiente de trabalho. Os retângulos vermelhos apresentam cada área discriminada como a seguir:

1 – Nesta área se define a seção de concreto, de forma poligonal, descritas como pontos num plano cartesiano  $xy$ , em centímetros. Não é necessário que a figura seja fechada (último ponto = primeiro ponto), pois o programa faz o processo automaticamente. Além disso, a posição do polígono em relação à origem (0,0) não importa, isto é, o centro de gravidade da seção pode estar posicionado em qualquer ponto.

2 – Nesta área o usuário preenche se a seção tem ou não armadura passiva. Caso haja armadura passiva, deve-se indicar a área de cada barra, e sua posição no mesmo plano cartesiano usado para se posicionar o polígono de seção de concreto;

3 – A área 3 é referente à presença e posição da armadura ativa na seção. Caso haja armadura ativa, o usuário deve fornecer sua posição no plano cartesiano  $xy$ , sua área, sua deformação de pré-alongamento (apesar de ser de tração, o usuário deve usar sinal positivo).

Caso o usuário deseje fazer a análise de ELS-W, é necessário fornecer o diâmetro da armadura também.

4 – O gráfico exposto na região 4 apresenta os resultados da análise de estado-limite último, isto é, o diagrama de momentos resistentes, e o gráfico da relação momento-curvatura. As outras 2 análises (Calcular Solicitações e Estados-limite de serviço) não utilizam de tal recurso para apresentar os seus resultados;

5 – A área 5 dispõe dados específicos às análises: por exemplo, na análise de estado-limite último, o usuário deve indicar qual a força normal atuante na seção; Cada área possui dados diferentes utilizados:

- Estado-limite último: Há um campo para se indicar a força normal na seção, e campos para apresentar momento atuante na seção; além disso, há o botão que inicia o processo de cálculo “Calcular”;

- Momento-Curvatura: Há o campo para indicar qual a força normal na seção, e o botão gatilho de cálculo “Calcular”;

- Verificar Solicitação de serviço: Neste ambiente, há três campos: força axial, momento em relação ao eixo  $x$  e momento em relação ao eixo  $y$ , além do botão que inicia o processo de cálculo “Calcular”;

- Calcular Solicitações: Nesta região há três campos: um para a deformação absoluta na fibra superior da seção  $\varepsilon_1$ , um para a deformação absoluta na fibra inferior da seção  $\varepsilon_2$ , e outro para o ângulo da linha-neutra  $\alpha$  em graus (sendo positivo no sentido horário).

Recapitulando, apresenta-se o citado na Figura A. 1:

Figura A. 1 – Ambiente de trabalho do programa OblqCALCO

The screenshot shows the OblqCALCO software interface with the following components:

- Top Menu:** Arquivo, Ajuda, Materiais.
- Left Panel:**
  - 1:** "Número de pontos: 4" with a table:
 

	X (cm)	Y (cm)
1	-10	-20
2	10	-20
3	10	20
4	-10	20
  - 2:** "Armadura Passiva" section with "Número Barras: 4" and a table:
 

	x (cm)	y (cm)	As (cm <sup>2</sup> )
1	-7	-17	1
2	7	-17	1
3	7	17	1
4	-7	17	1
  - 3:** "Armadura Ativa" section with "Número Barras: 2" and a table:
 

	x (cm)	y (cm)	Ap (cm <sup>2</sup> )	sp (%)	Φ (mm)
1	-7	-17	1	5	0
2	7	-17	1	5	0
  - 4:** A grid plot with x and y axes ranging from -1 to 1.
  - 5:** "Calculo Solicitações" section with input fields for:
    - Nd (kN): 0
    - Mxx (kN.cm): 0
    - Myy (kN.cm): 0
    - Nmax (kN): 0
    - Nmin (kN): 0
 Status: Ocioso. A "Calcular" button is present.
- Right Panel:** "Salvar Resultados..." button.
- Bottom Status Bar:**
  - fck = 30 MPa, fcd = 2,1429 kN/cm<sup>2</sup>, n = 2, Deformações c2 e cu = 2 e 3,5
  - Fyk = 500 MPa, Fyd = 43,478 kN/cm<sup>2</sup>, Es = 21000 kN/cm<sup>2</sup>, Deformação Limite do Aço = 2,0704

Há 2 janelas de controle secundário: uma para materiais e uma para os cálculos:

Na janela de materiais, apresentada pela figura a seguir, o usuário escolhe as características do concreto, aço de armadura passiva e aço da armadura ativa. O significado de cada é explicado na sequência, iniciando pela janela de materiais, apresentada na Figura A. 2.

Figura A. 2 – Ambiente de configuração dos materiais no programa OblqCALCO

Propriedade	Valor
$f_{ck}$ (MPa)	30
$f_{yk}$ (MPa)	500
$f_{pyd}$ (MPa)	1460
$\gamma_c$	1,4
$\gamma_s$	1,15
$f_{ptd}$ (MPa)	1626
$E_s$ (GPa)	210
$E_p$ (GPa)	200
$\epsilon_{su}$ (1/1000)	10
$\epsilon_{p,u}$ (1/1000)	35
$\eta_1$	1,2

- $f_{ck}$  : resistência característica à compressão do concreto;
- $\gamma_c$  : coeficiente que relaciona  $f_{ck}$  com  $f_{cd}$  (resistência de cálculo à compressão do concreto), de padrão 1,4;
  - $f_{yk}$  : tensão de escoamento característica do aço de armadura passiva: na norma, sob nomenclatura  $E_{cs}$ , ver figura 8.4 da norma ABNT NBR 6118:2014;
- $f_{yd}$  : tensão de escoamento de cálculo do aço de armadura passiva: na norma, sob nomenclatura  $E_{cs}$ , ver figura 8.4 da norma ABNT NBR 6118:2014;
- $E_s$  : módulo de elasticidade do aço de armadura passiva: na norma, sob nomenclatura  $E_{cs}$ , ver figura 8.4 da norma ABNT NBR 6118:2014 ;
- $\epsilon_{su}$ : deformação última do aço (cujo valor padrão é 10 ‰) figura 8.4 da norma ABNT NBR 6118:2014 ;
- $f_{pyd}$ : resistência de cálculo ao escoamento de escoamento da armadura de protensão, ver figura 8.5 da norma ABNT NBR 6118:2014;
- $f_{ptd}$ : resistência de cálculo à tração da armadura de protensão, ver figura 8.5 da norma ABNT NBR 6118:2014 ;
- $E_p$ : módulo de elasticidade do aço da armadura ativa (cujo valor padrão é 200 GPa) figura 8.5 da norma ABNT NBR 6118:2014;
- $\epsilon_{pu}$ : deformação última do aço de protensão (cujo valor padrão é 35 ‰) figura 8.5 da norma ABNT NBR 6118:2014;

Na janela de configurações, o usuário escolhe os critérios de parada para os processos iterativos:  $(\varepsilon_i - \varepsilon_{i-1})$  quer dizer a diferença entre uma iteração e a próxima, que são os critérios de parada usados no cálculo numérico.

Também é relevante conhecer os outros dois campos que não fazem parte do critério de parada do processo iterativo:

- “*Critérios ELU –  $\alpha$* ”: um diagrama de Flexão Oblíqua Composta é uma curva: mas como ele é aproximado, neste trabalho essa curva é trocada por vários pontos: um ponto para cada ângulo que a linha-neutra girar em relação à seção a cada iteração concluída:  $\alpha$  é exatamente a distância entre esses ângulos.

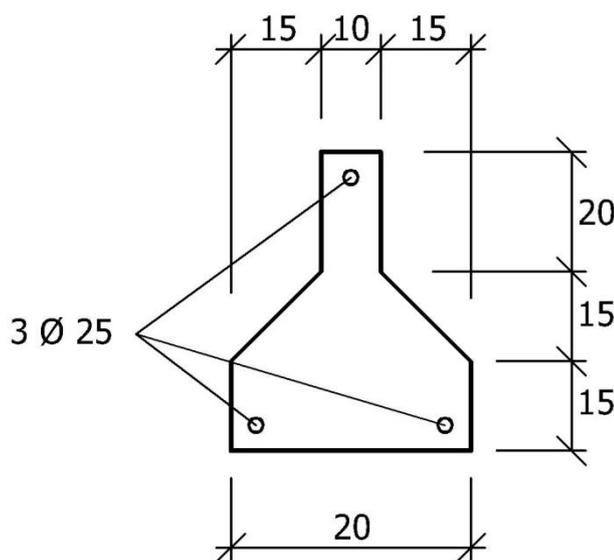
- “*Critérios ELS-W, ELS-F e ELS-D –  $\varepsilon_{ELS-F}$* ”: o usuário pode escolher a deformação limite para o estado-limite de formação de fissuras. Sugere-se que se consultar a seção que trata deste assunto na dissertação completa e o tópico da norma ABNT NBR 6118:2014. Além disso, o programa não suporta considerar tensões de tração no concreto.

## A.2 ENTRADA DE DADOS

A entrada de dados é realizada pela entrada das coordenadas cartesianas que definem a seção.

Apresenta-se como tal entrada deve ser realizada mostrando uma seção exemplo, cuja representação esquemática é apresentada na Figura A. 3. Essa mesma seção será a seção avaliada em todos os ambientes do programa. É composta de concreto C30 e armadura passiva CA 50. Não é utilizada a armadura ativa.

Figura A. 3 – Seção exemplo de entrada no programa OblqCALCO



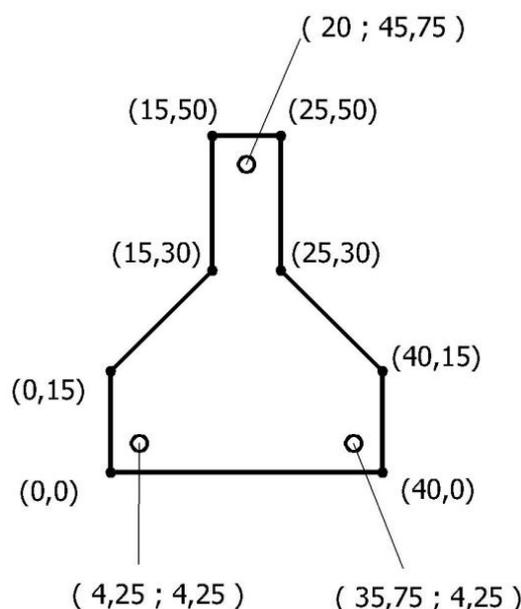
Para isso, aplica-se no programa os dados:

- Polígono que define a seção de concreto: resistência característica à compressão do concreto: foram aplicados os pontos:  $(0, 0)$ ;  $(40, 0)$ ;  $(40, 15)$ ;  $(25, 30)$ ;  $(25, 50)$ ;  $(15, 50)$ ;  $(15, 30)$ ;  $(0, 15)$ .

- Número de barras da armadura e as coordenadas de seus centros:  $(4,25; 4,25)$ ;  $(35,75; 4,25)$ ;  $(20; 45,75)$  com seus respectivas áreas que são iguais nesse caso:  $4,9 \text{ cm}^2$ .

Na Figura A. 4 constam as coordenadas de cada ponto, para a seção de concreto e para a armadura passiva.

Figura A. 4 – Coordenadas dos pontos da seção exemplo.



### A.3 EXPORTAÇÃO DE DADOS

Outra utilidade que o programa pode ter é permitir que os dados sejam exportados de alguma maneira. Isso permite que outros usuários possam comparar resultados: pois com o acesso a um gráfico, não se conhece exatamente os valores apresentados. Porém, com acesso aos valores, pode-se gerar gráficos que comparem diferentes análises.

O programa não gera arquivos, ou tabelas prontas, mas exporta os dados num campo de texto. Neste campo, cada análise está numa linha, para que o usuário possa facilmente “Colar” os dados numa planilha. Alguns caracteres não são aceitos por certos programas. A seguir é apresentada uma linha do campo de texto: “ $Eps1=0; Eps2=0; força=0; Curv.=0 \times 10/m; Mxx = 0; Myy = 0$ ”. Neste exemplo, o usuário pode, por exemplo, passar por um gerenciador de texto, como o *Bloco de Notas*, por exemplo, e aplicar a ferramenta “Substituir” para trocar os “x” por *nada*, eliminando assim caracteres indesejados. Os programas de

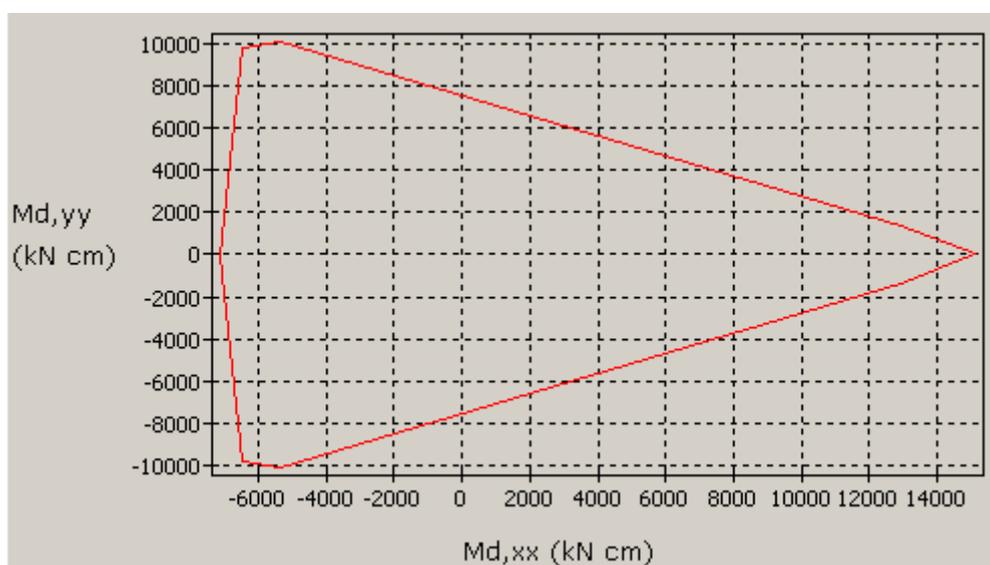
planilha também transformam texto em tabela, para facilitar essa utilidade. Na seção referente ao ELU a ultima imagem foi gerada com uso dessa utilidade.

#### A.4 ABA “ELU”

Pode-se obter diagramas de estado-limite último de uma seção específica, apresentando os momentos resistentes  $M_{xx}$  e  $M_{yy}$  segundo uma força axial desejada, por exemplo, deseja-se verificar a seguinte seção, com materiais concreto C30 e aço CA-50.

Como resultado, o programa apresenta o gráfico e os dados num painel. O gráfico é apresentado como mostra a Figura A. 5:

Figura A. 5 – Resultado gráfico da análise de estado-limite último da seção exemplo.



O programa também extrai as duplas de pontos ( $M_{xx}$  e  $M_{yy}$ ) sob um campo de texto, como apresenta a Figura A. 6 a seguir.

Figura A. 6 – Resultado da análise de estado-limite último da seção exemplo como texto.

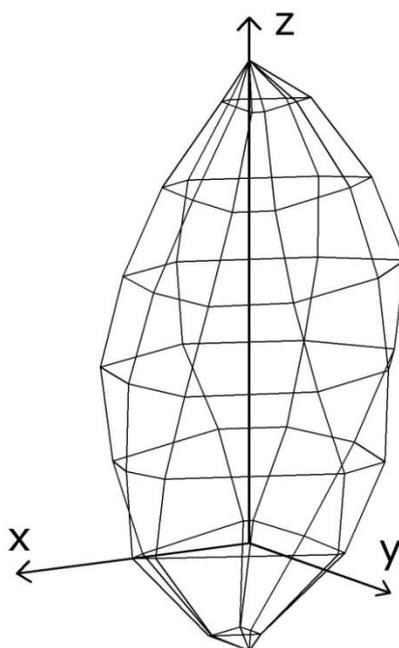
```

Angulo=324; Eps1=3,5; Eps2=-9,0165; x/h=0,27963; força=4,9253E-7; Mxx = 14003; Myy =
10037
Angulo=327; Eps1=3,5; Eps2=-9,0584; x/h=0,2787; força=1,1349E-6; Mxx = 14524; Myy =
9314,4
Angulo=330; Eps1=3,5; Eps2=-9,0186; x/h=0,27958; força=1,0551E-7; Mxx = 14994; Myy =
8555,7
Angulo=333; Eps1=3,5; Eps2=-8,9794; x/h=0,28046; força=9,7405E-7; Mxx = 15424; Myy =
7772,1
Angulo=336; Eps1=3,5; Eps2=-8,9404; x/h=0,28134; força=-1,778E-7; Mxx = 15812; Myy =
6965,9
Angulo=339; Eps1=3,5; Eps2=-8,9015; x/h=0,28222; força=-2,9706E-7; Mxx = 16157; Myy =
6139,7
Angulo=342; Eps1=3,5; Eps2=-8,8625; x/h=0,28311; força=3,6015E-7; Mxx = 16458; Myy =
5295,8
Angulo=345; Eps1=3,5; Eps2=-8,8231; x/h=0,28402; força=-6,6132E-8; Mxx = 16715; Myy =
4436,7
Angulo=348; Eps1=3,5; Eps2=-8,783; x/h=0,28495; força=1,0196E-6; Mxx = 16926; Myy =
3564,8
Angulo=351; Eps1=3,5; Eps2=-8,7422; x/h=0,2859; força=1,2757E-6; Mxx = 17092; Myy =
2682,7
Angulo=354; Eps1=3,5; Eps2=-8,7002; x/h=0,28688; força=2,3175E-7; Mxx = 17210; Myy =
1792,9
Angulo=357; Eps1=3,5; Eps2=-8,6569; x/h=0,2879; força=1,2346E-7; Mxx = 17283; Myy =
897,77
Angulo=360; Eps1=3,5; Eps2=-8,6119; x/h=0,28897; força=-5,8974E-7; Mxx = 17308; Myy = 0

```

Esse texto pode ser extraído usando o botão “Salvar Resultados...”: assim o usuário pode facilmente desenhar o gráfico em quaisquer outros lugares. Foi elaborado um diagrama com dados de várias iterações para a seção padrão, apresentado na Figura A. 7. Para elaborar o diagrama, adotou-se um ângulo de  $45^\circ$  entre os resultados, e as forças normais de cada patamar é de -500, 0, +500, +1000, +1500, +2000, +2500 e os pontos extremos de forças axiais. Foi usada uma escala de 20 na força axial para tornar o entendimento mais intuitivo.

Figura A. 7 – Envoltória de momentos resistentes da seção exemplo.



Inclusive, sugere-se para trabalhos futuros o desenvolvimento de uma forma automática de desenhar os diagramas mostrados, usando programação LISP em conjunto com o programa mostrado, por exemplo. Para mais informações sugere-se a leitura da dissertação relativa ao programa.

#### A.5 ABA “MOMENTO-CURVATURA”

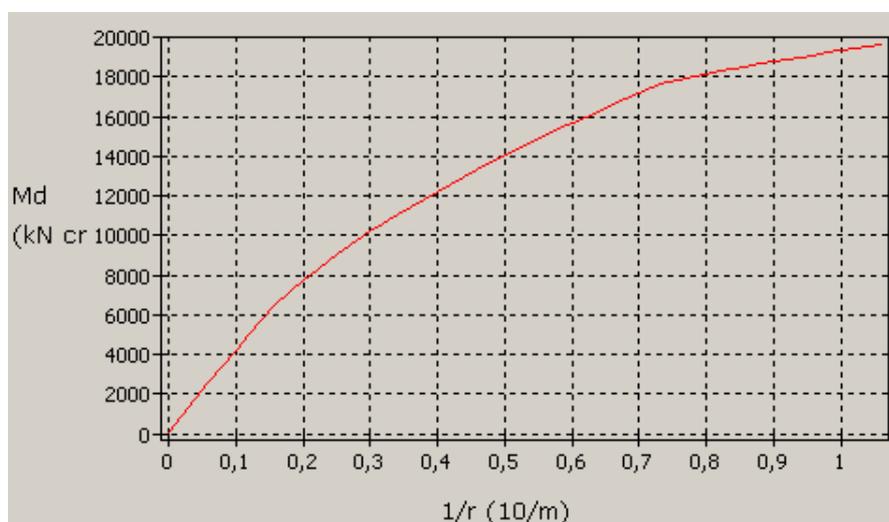
Este ambiente foi elaborado com a intenção de descrever a relação sugerida pelo título: a relação momento-curvatura, segundo uma força axial constante. O programa primeiro gira a seção segundo o ângulo desejado, e depois calcula a posição da linha neutra que tem a força axial e curvatura correspondente e registra o valor. O programa então calcula os momentos  $M_{xx}$  e  $M_{yy}$  e os apresenta no campo de texto. Além disso, o programa gera um gráfico de  $M_{xx} \times 1/r$ .

Por que o programa não apresenta um gráfico “ $M \times 1/r$ ”? Pois um gráfico assim levaria o usuário à falsa impressão que o ângulo de inclinação da linha-neutra na seção é igual ao ângulo entre o vetor  $M$  e o eixo  $y$ . Ou ainda a falsa impressão de  $M_{xx}$  e  $M_{yy}$  manterem relação linear entre si.

Por esse motivo o programa avisa quando houver momento nas duas direções: para prevenir o usuário de usar dados de momento incorretamente ou sem devido aviso: evitando que a resistência à momentos numa direção seja superestimada.

Finalmente, apresenta-se um gráfico obtido pelo programa para a relação momento-curvatura na Figura A. 8. Esse gráfico é apresentado para a seção exemplo deste texto apresentada no título A.2. Além disso, a força axial é de 500 kN.

Figura A. 8 – Resultado gráfico da análise momento-curvatura para a seção exemplo.

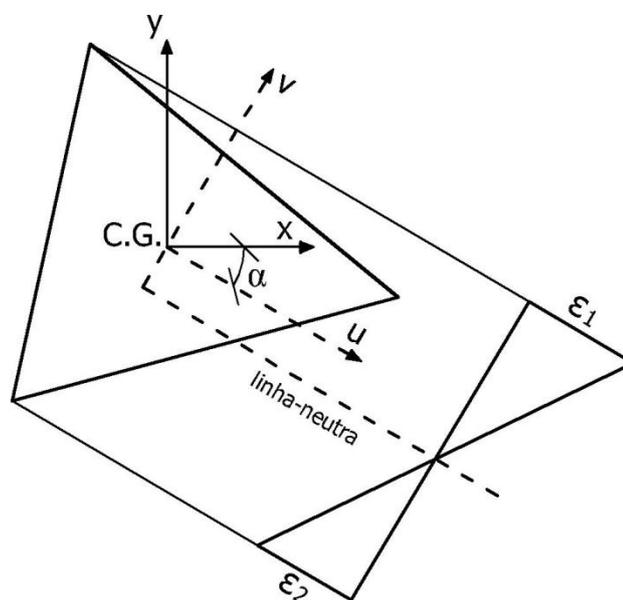


## A.6 ABA “CALCULAR SOLICITAÇÕES”

A aba calcular solicitações tem por objetivo calcular diretamente os esforços análogos a certa conformação de deformações. O usuário fornece as deformações da seção usando as variáveis  $\alpha$ ,  $\varepsilon_1$  e  $\varepsilon_2$ , como indica a figura logo após a descrição de cada variável:

- $\alpha$ : é o ângulo de inclinação da linha-neutra;
- $\varepsilon_1$ : é a deformação no ponto da seção com a maior deformação (ou ponto com extremo máximo em relação ao eixo  $uv$ ).
- $\varepsilon_2$ : é a deformação no ponto da seção com a menor deformação (ou ponto com extremo mínimo em relação ao eixo  $uv$ ).

Figura A. 9 – Variáveis utilizadas no ambiente “Calcular Solicitações”.



O programa apenas resolve situações em que a borda do extremo máximo (maior compressão na seção) está mais comprimida que a borda do extremo mínimo (maior tração na seção). Caso for necessário resolver uma situação inversa (tração nas bordas superiores e compressão nas bordas inferiores – sendo superior e inferior em relação ao eixo  $uv$ ), então inverte-se a seção por  $180^\circ$ . O programa faz isso automaticamente e avisa o usuário.

Então, o programa retorna como resultado a força axial, momento em relação ao eixo  $x$  e momento em relação ao eixo  $y$ . E também, o ângulo do vetor momento fletor: sendo que  $0^\circ$  significa um momento em relação ao eixo  $x$  convencional para vigas sob carregamento apenas de origem gravitacional.

A principal utilidade dessa aba é possibilitar ao usuário estudar o comportamento das seções ao alcançar certos valores de deformações. Além disso, a aba pode ser usada para conferir o funcionamento do programa, principalmente para comparação de resultados de

diferentes rotinas, pois muitos programas de análise de Flexão Oblíqua Composta consideram forças e momentos como função das deformações, e não o contrário.

#### A.7 ABA “VERIFICAR SOLICITAÇÕES - ELS”

A quarta e última aba do programa faz com que o programa procure a conformação de deformações que apresentam certa solicitação. Isto é, o usuário escolhe as solicitações (força axial, momento fletor em relação ao eixo  $x$ , momento fletor em relação ao eixo  $y$ ) e o programa procura pela conformação de deformações, retornando o resultado nas variáveis  $\alpha$ ,  $\varepsilon_1$  e  $\varepsilon_2$ .

Além disso, o programa auxilia o usuário para verificações dos estados-limites de serviço de abertura de fissuras (ELS-W), formação de fissuras (ELS-F) e descompressão (ELS-D).

Salienta-se que a resistência à tração do concreto não é considerada em qualquer análise, mesmo aquelas em que o concreto não fissura (ELS-F e ELS-D). Além disso, o critério usado para analisar o ELS-F não é levado em conta de maneira idêntica à norma, mas usando uma aproximação: na norma, sugere-se realizar um cálculo que leva em conta a força da seção, sua inércia, e assim obter o momento de fissuração, como incida a seção 17.3 da norma ABNT NBR 6118:2014.

Neste trabalho levou-se em conta a relação tensão-deformação do concreto apenas. Na seção 8.2 a mesma norma caracteriza o concreto estrutural, e apresenta um gráfico que mostra o comportamento à tração do concreto, que indica a ruptura ao atingir deformação de 0,15‰ à tração.

- ELS-W: O programa calcula a tensão de cada armadura, e calcula a equação da norma da abertura de fissuras. A equação da norma é o menor valor entre 2 expressões. No presente trabalho, apenas se calcula a primeira (que não considera o concreto no envolvimento da armadura). Mesmo assim, se fornece a tensão, caso o usuário deseje calculá-la.

A equação que o programa calcula é:

$$w_{k,1} = \frac{\phi_i}{12,5\eta_1} \frac{\sigma_{si}}{E_{si}} \frac{3\sigma_{si}}{f_{ctm}}$$

- ELS-F: Para essa verificação, o programa compara a deformação de maior tração na seção (isto é, menor deformação, visto que deformação negativa = alargamento; deformação positiva = encurtamento) com o a máxima deformação permitida (limite de ruptura à tração

do concreto, cujo padrão é 0,15‰ ). Caso nenhuma deformação na seção seja menor que 0,15‰ o ELS-F é atendido.

ELS-D: Funciona da mesma do ELS-F, porém com limites diferentes: no limite de descompressão, nenhum ponto da seção apresenta tração. Logo, a menor deformação permitida é *zero*. Logo, compara-se a menor deformação na seção com *zero*. Caso nenhuma deformação na seção seja menor que *zero*, o ELS-D é atendido.

## A.8 USO DE PROTENSÃO

O programa tem capacidade de avaliar as seções com presença de armadura ativa. Para isso, o usuário precisa incluir dois dados mais que a armadura passiva (além de caracterizar os materiais):

-  $\varepsilon_p$  : a deformação de pré alongamento  $\varepsilon_p$  em 1/1000. Para calculá-la, o usuário deve calcular a força de protensão com a consideração das perdas, e calcular qual a deformação correspondente. Essa deformação é de alargamento nos cabos, mesmo assim, seu valor no programa deve ser positivo.

-  $\Phi$ (mm): diâmetro da seção da armadura. É o diâmetro da armadura, APENAS para cálculo do ELS-W. Isso é necessário para possibilitar várias considerações, por exemplo calcular o diâmetro a partir da área (usando a fórmula de áreas de circunferências) ou usar o diâmetro total da cordoalha. Caso a análise em ELS-W não seja importante, o usuário pode preencher com “zeros”. NÃO devendo deixar a tabela com vazios.

A força de protensão ocasiona a seção à forças de compressão. O usuário não precisa considera-las separadamente, o programa faz isso automaticamente. A seguir, um exemplo de inserção da armadura  $\Phi 12,7$ , com  $\varepsilon_p = 6‰$  (tração).

Figura A. 10 – Entrada de dados da armadura ativa.

Armadura Ativa  
 Número Barras: 2

	x (cm)	y (cm)	As (cm <sup>2</sup> )	$\varepsilon_p$ (‰)	$\Phi$ (mm)
1	0	0	1	6	12,7
2	0	-15	1	6	12,7

## B. APÊNDICE 2 – CÓDIGOS DO PROGRAMA OBLQCALCO

### B.1 ROTINA DE ESTADO-LIMITE ÚLTIMO

A rotina de estado-limite último tem como objetivo final determinar vários valores de momentos resistentes da seção, construindo uma envoltória. Apresenta-se a rotina elaborada neste trabalho:

```

procedure TFrmMainFrm.ButGoELU(Sender: TObject);
var
iContador:integer;
fpdEpsilC2,fpEpsilonNow,fpEpsilonPrevious,fpNNow,fpNPrevious,fpNAuxiliar,fp
EpsilonAuxiliar
,fpNretaA,fpNdominio23,fpNdominio45,fpNretaB:real;

begin

if fVarBin=True then
begin
LabelStatusELU.Caption:='Status: Calculando... Aguarde';
LabelStatusELU.Refresh;

// Inicio da preparação para iteração
falfatemp:=0;
Serie1.Clear;
SeriePonto.Clear;
MemoELU.Clear;
ProgressBarELU.Position:=0;
LabelChartX.Caption:='Md,xx (kN cm)';
LabelChartYSup.Caption:='Md,yy';
LabelChartYInf.Caption:='(kN.cm)';

repeat
RotacaoDePontos(falfatemp,CoordSecT,CoordSec);
if fbArmPas then RotacaoDePontos(falfatemp,CoordArmPasT,CoordArmPas);
if fbArmAtiv then
RotacaoDePontos(falfatemp,CoordArmAtivT,CoordArmAtiv);
PontosMaximoMinimo(CoordSec,fvmax,fvmin);
fhalfa:=fvmax-fvmin;
if fbArmPas then PontosMaximoMinimo(CoordArmPas,fAvmax,fAvmin);
if fbArmPas then fdalfa:=fvmax-fAvmin;

// PRIMEIRO, CALCULA-SE EM QUAL REGIAO ESTA A FORCA NORMAL DESEJADA
// N RETA A - LIMITE DOMINIO 1 - Nem se calcula a força no concreto.
fEpsilon1:=fEpsilonSu;
fEpsilon2:=fEpsilonSu;
fpNretaA:=0;

if fbArmPas then begin
for fTemp:=0 to Length(CoordArmPasT)-1 do begin // Inicio força do
aço
ahsi[fTemp]:=fvmax-CoordArmPas[fTemp].yi;

```

```

        aEpsilonSi[fTemp]:=fEpsilon1-ahsi[fTemp]*(fEpsilon1-
fEpsilon2)/fhalfa;

fpNretaA:=fpNretaA+aAs[fTemp]*TensaoArmPas(aEpsilonSi[fTemp],fEs,ffyd);
    end; //fim força do aço
    end;

    if fbArmAtiv then begin
        for fTemp:=0 to Length(CoordArmAtivT)-1 do begin // Inicio força do
aço
            ahpi[fTemp]:=fvmax-CoordArmAtiv[fTemp].yi;
            aEpsilonPsi[fTemp]:=fEpsilon1-ahpi[fTemp]*(fEpsilon1-
fEpsilon2)/fhalfa;
            varTemp:=aEpsilonPsi[fTemp]+aEpsilonPrealong[fTemp];

fpNretaA:=fpNretaA+aAp[fTemp]*TensaoArmAtiv(fEpsilonpyd,fEpsilonpu,ffpyd,ff
ptd,varTemp);
        end; //fim força do aço
        end;
        // N DOMINIO 23
        fEpsilon1:=fEpsilonCU;
        fEpsilon2:=fEpsilonSU; // ESSE VALOR DE EPSILON2 É USADO CASO A RUPTURA
SEJA NO CONCRETO
        if fModoDeRuptura=1 then fEpsilon2:=fEpsilon1-(fEpsilon1-
fEpsilonSU)*fhalfa/fdalfa; // SE FOR NA ARMADURA PASSIVA, ESSE VALOR
        if fEpsilon1=fEpsilon2 then begin
            fxEpsilonC2:=10e6;
            fxalfa:=fxEpsilonC2*fEpsilon1/fEpsilonC2;
        end;
        if fEpsilon1<>fEpsilon2 then begin
            fxalfa:=fhalfa*fEpsilon1/(fEpsilon1-fEpsilon2); //Explicacoes na
secao 4.1.2;
            fxEpsilonC2:=fhalfa*fEpsilonC2/(fEpsilon1-fEpsilon2); //Explicacoes
na secao 4.1.2;
        end;

fpNdominio23:=ForcaConcreto(CoordSec,ffcd,fxalfa,fxEpsilonC2,fNTensaoConcre
to);

    if fbArmPas then begin
        for fTemp:=0 to Length(CoordArmPasT)-1 do begin // Inicio força do
aço
            ahsi[fTemp]:=fvmax-CoordArmPas[fTemp].yi;
            aEpsilonSi[fTemp]:=fEpsilon1-ahsi[fTemp]*(fEpsilon1-
fEpsilon2)/fhalfa;

fpNdominio23:=fpNdominio23+aAs[fTemp]*TensaoArmPas(aEpsilonSi[fTemp],fEs,ff
yd);
        end; //fim força do aço
        end;
        if fbArmAtiv then begin
            for fTemp:=0 to Length(CoordArmAtivT)-1 do begin // Inicio força do
aço
                ahpi[fTemp]:=fvmax-CoordArmAtiv[fTemp].yi;
                aEpsilonPsi[fTemp]:=fEpsilon1-ahpi[fTemp]*(fEpsilon1-
fEpsilon2)/fhalfa;
                varTemp:=aEpsilonPsi[fTemp]+aEpsilonPrealong[fTemp];

fpNdominio23:=fpNdominio23+aAp[fTemp]*TensaoArmAtiv(fEpsilonpyd,fEpsilonpu,
ffpyd,ffptd,varTemp);
            end;
        end;
    end;

```

```

    end; //fim força do aço
end;
// N DOMINIO 45
fEpsilon1:=fEpsilonCU;
fEpsilon2:=0;

if fEpsilon1=fEpsilon2 then begin
    fxEpsilonC2:=10e6;
    fxalfa:=fxEpsilonC2*fEpsilon1/fEpsilonC2;
end;
if fEpsilon1<>fEpsilon2 then begin
    fxalfa:=fhalfa*fEpsilon1/(fEpsilon1-fEpsilon2);
    fxEpsilonC2:=fhalfa*fEpsilonC2/(fEpsilon1-fEpsilon2); //Explicacoes
na secao 4.1.2;
end;

fpNdominio45:=ForcaConcreto(CoordSec,ffcd,fxalfa,fxEpsilonC2,fNTensaoConcreto);

if fbArmPas then
begin
    for fTemp:=0 to Length(CoordArmPasT)-1 do // Inicio força do aço
        begin
            ahSi[fTemp]:=fvmax-CoordArmPas[fTemp].yi;
            aEpsilonSi[fTemp]:=fEpsilon1-ahSi[fTemp]*(fEpsilon1-
fEpsilon2)/fhalfa;

fpNdominio45:=fpNdominio45+aAs[fTemp]*TensaoArmPas(aEpsilonSi[fTemp],fEs,ff
yd);

            end;
        end; //fim força do aço
    if fbArmAtiv then
        begin
            for fTemp:=0 to Length(CoordArmAtivT)-1 do // Inicio força do aço
                begin
                    ahPi[fTemp]:=fvmax-CoordArmAtiv[fTemp].yi;
                    aEpsilonPsi[fTemp]:=fEpsilon1-ahPi[fTemp]*(fEpsilon1-
fEpsilon2)/fhalfa;
                    varTemp:=aEpsilonPsi[fTemp]+aEpsilonPrealong[fTemp];

fpNdominio45:=fpNdominio45+aAp[fTemp]*TensaoArmAtiv(fEpsilonpyd,fEpsilonpu,
ffpyd,ffptd,varTemp);

                    end; //fim força do aço
                end;

// N REBA B - LIMITE DOMINIO 5
fEpsilon1:=fEpsilonC2;
fEpsilon2:=fEpsilonC2;

if fEpsilon1=fEpsilon2 then begin
    fxEpsilonC2:=10e6;
    fxalfa:=fxEpsilonC2*fEpsilon1/fEpsilonC2;
end;
if fEpsilon1<>fEpsilon2 then begin
    fxalfa:=fhalfa*fEpsilon1/(fEpsilon1-fEpsilon2);
    fxEpsilonC2:=fhalfa*fEpsilonC2/(fEpsilon1-fEpsilon2); //Explicacoes
na secao 4.1.2;
end;

```

```

fpNretaB:=ForcaConcreto(CoordSec,ffcd,fxalfa,fxEpsilonC2,fNTensaoConcreto);

  if fbArmPas then begin
    for fTemp:=0 to Length(CoordArmPasT)-1 do begin // Inicio força do
aço
      ahsi[fTemp]:=fvmax-CoordArmPas[fTemp].yi;
      aEpsilon1:=fEpsilon1-ahsi[fTemp]*(fEpsilon1-
fEpsilon2)/fhalfa;
fpNretaB:=fpNretaB+aAs[fTemp]*TensaoArmPas(aEpsilon1[fTemp],fEs,ffyd);
      end; //fim força do aço
    end;
    if fbArmAtiv then begin
      for fTemp:=0 to Length(CoordArmAtivT)-1 do begin // Inicio força do
aço
        ahpi[fTemp]:=fvmax-CoordArmAtiv[fTemp].yi;
        aEpsilonpsi[fTemp]:=fEpsilon1-ahpi[fTemp]*(fEpsilon1-
fEpsilon2)/fhalfa;
        varTemp:=aEpsilonpsi[fTemp]+aEpsilonprealong[fTemp];
fpNretaB:=fpNretaB+aAp[fTemp]*TensaoArmAtiv(fEpsilonpyd,fEpsilonpu,ffpyd,ff
ptd,varTemp);
        end; //fim força do aço
      end;

      if fNdFixado<=fpNretaB then begin
        fVarredura:=3; //Regiao I - Epsilon 2 usado como referencia - pois se
Concreto for C90, Epsilon1 é fixo
        fpNPrevious:=fpNdominio45;
        fpEpsilonPrevious:=0;
        fpNNow:=fpNretaB;
        fpEpsilonNow:=fEpsilonC2;
        fpdEpsilC2:=(fEpsilonCU-fEpsilonC2)/fEpsilonCU*fhalfa;
        end;
      if fNdFixado<=fpNdominio45 then begin
        fVarredura:=2; //Regiao II - Epsilon 2 usado como referencia -
Epsilon1 assume valor fixo
        fpNPrevious:=fpNdominio23;
        if fModoDeRuptura=0 then fpEpsilonPrevious:=fEpsilonSU;
        if fModoDeRuptura=1 then fpEpsilonPrevious:=fEpsilonCU-(fEpsilonCU-
fEpsilonSU)*fhalfa/fdalfa;
        fpNNow:=fpNdominio45;
        fpEpsilonNow:=0;
        fEpsilon1:=fEpsilonCU;
        fEpsilon2:=0;
        end;
      if fNdFixado<=fpNdominio23 then begin
        fVarredura:=1; //Regiao III - Epsilon 1 usado como referencia -
Epsilon 2 fixo
        fpNPrevious:=fpNretaA;
        fpEpsilonPrevious:=-fEpsilonYD;
        fpNNow:=fpNdominio23;
        fpEpsilonNow:=fEpsilonCU;
        if fModoDeRuptura=0 then fEpsilon2:=fEpsilonSU;
        if fModoDeRuptura=1 then fEpsilon2:=fEpsilonCU-(fEpsilonCU-
fEpsilonSU)*fhalfa/fdalfa;
        end;
      if fNdFixado<=fpNretaA then begin
        fVarredura:=0; //Alem do limite
      end;
    end;
  end;

```

```

while fVarredura<>0 do begin
  if fVarredura=3 then begin // Regiao I - Eps2 referencia - Eps1
funcao de Eps2
    fpEpsilonNow:=fpEpsilonNow-(fpEpsilonNow-
fpEpsilonPrevious)/(fpNNow-fNPrevious)*(fpNNow-fNdFixado);
    if fpEpsilonNow<0 then fpEpsilonNow:=0-(0-
fpEpsilonPrevious)/(fpNdominio45-fNPrevious)*(fpNdominio45-fNdFixado);
    if fpEpsilonNow>fEpsilonC2 then fpEpsilonNow:=fEpsilonC2-
(fEpsilonC2-fpEpsilonPrevious)/(fpNretaB-fNPrevious)*(fpNretaB-fNdFixado);
    fpEpsilonPrevious:=fpEpsilon2;
    fpNPrevious:=fpNNow;
    fEpsilon2:=fpEpsilonNow;
    fEpsilon1:=(fEpsilonC2*fEpsilonCU-(fEpsilonCU-
fEpsilonC2)*fEpsilon2)/fEpsilonC2;
  end;
  if fVarredura=2 then begin // Regiao II
    fpEpsilonNow:=fpEpsilonNow-(fpEpsilonNow-
fpEpsilonPrevious)/(fpNNow-fNPrevious)*(fpNNow-fNdFixado);
    if fModoDeRuptura=0 then begin
      if fpEpsilonNow<fEpsilonsu then begin
        fpEpsilonNow:=fEpsilonsu;
        fpNNow:=fpNdominio23;
      end;
    end;
    if fModoDeRuptura=1 then begin
      if fpEpsilonNow<(fEpsilonCU-(fEpsilonCU-
fEpsilonsu)*fhalfa/fdalfa) then fpEpsilonNow:=fEpsilonsu-(fEpsilonsu-
fpEpsilonPrevious)/(fpNdominio23-fNPrevious)*(fpNdominio23-fNdFixado);
    end;
    if fpEpsilonNow>0 then fpEpsilonNow:=0-(0-
fEpsilon2)/(fpNdominio45-fNNow)*(fpNdominio45-fNdFixado); // Isto é, se
sair do intervalo para um valor maior, compara-se o epsilon com o valor
maximo
    fpEpsilonPrevious:=fEpsilon2;
    fpNPrevious:=fpNNow;
    fEpsilon2:=fpEpsilonNow;
  end;
  if fVarredura=1 then begin // Regiao III
    if fpNNow<=fpNretaA then begin
      fpNPrevious:=fpNdominio23;
      fpEpsilonPrevious:=fEpsilonCU;
    end;
    if abs(fpEpsilonNow-fpEpsilonPrevious)<=0.000000001 then begin
      fpNPrevious:=fpNdominio23;
      fpEpsilonPrevious:=fEpsilonCU;
    end;
    fpEpsilonNow:=fpEpsilonNow-(fpEpsilonNow-
fpEpsilonPrevious)/(fpNNow-fNPrevious)*(fpNNow-fNdFixado);
    if fpEpsilonNow<fEpsilonsu then fpEpsilonNow:=fEpsilonsu;
    if fpEpsilonNow>fEpsilonCU then fpEpsilonNow:=fEpsilonCU;
    fpEpsilonPrevious:=fEpsilon1;
    fpNPrevious:=fpNNow;
    fEpsilon1:=fpEpsilonNow;
    if fModoDeRuptura=1 then fEpsilon2:=fEpsilon1-(fEpsilon1-
fEpsilonsu)*fhalfa/fdalfa;
    // Caso o modo de ruptura seja no concreto, o valor é constante
fEpsilonsu
  end;

  if abs(fpEpsilonNow-fpEpsilonPrevious)<=fEpsilonErroELU then begin
    fVarredura:=0;
  end;
end;

```

```

end;

// Arredondar os epsilon, pois eles estão adquirindo erros (nas
casas decimais avançadas
fStrtemp:=FloattostrF(fEpsilon1,ffGeneral,8,6);
fEpsilon1:=StrToFloat(fStrtemp);
fStrtemp:=FloattostrF(fEpsilon2,ffGeneral,8,6);
fEpsilon2:=StrToFloat(fStrtemp);
if abs(fEpsilon1)<0.000000001 then fEpsilon1:=0;
if abs(fEpsilon2)<0.000000001 then fEpsilon2:=0;

if fEpsilon1=fEpsilon2 then begin
fxEpsilonC2:=10e6;
fxalfa:=fxEpsilonC2*fEpsilon1/fEpsilonC2;
end;
if fEpsilon1<>fEpsilon2 then begin
fxalfa:=fhalfa*fEpsilon1/(fEpsilon1-fEpsilon2);
fxEpsilonC2:=fhalfa*fEpsilonC2/(fEpsilon1-fEpsilon2);
//Explicacoes na secao 4.1.2;
end;

fpNNow:=ForcaConcreto(CoordSec,ffcd,fxalfa,fxEpsilonC2,fNTensaoConcreto);

if fbArmPas then
begin
for fTemp:=0 to Length(CoordArmPasT)-1 do // Inicio
força do aço
begin
ahsi[fTemp]:=fvmax-CoordArmPas[fTemp].yi;
aEpsilonpsi[fTemp]:=fEpsilon1-
ahsi[fTemp]*(fEpsilon1-fEpsilon2)/fhalfa;
fpNNow:=fpNNow+aAs[fTemp]*TensaoArmPas(aEpsilonpsi[fTemp],fEs,ffyd);
end; //fim força do aço
end;

if fbArmAtiv then
begin
for fTemp:=0 to Length(CoordArmAtivT)-1 do // Inicio força
da armadura passiva
begin
ahpi[fTemp]:=fvmax-CoordArmAtiv[fTemp].yi;
aEpsilonpsi[fTemp]:=fEpsilon1-
ahpi[fTemp]*(fEpsilon1-fEpsilon2)/fhalfa;
varTemp:=aEpsilonpsi[fTemp]+aEpsilonprealong[fTemp];
fpNNow:=fpNNow+aAp[fTemp]*TensaoArmAtiv(fEpsilonpyd,fEpsilonpu,ffpyd,ffptd,
varTemp);
end; //fim força da armadura passiva
end;

if fVarredura=0 then // Inicio do calculo dos momentos - se a
carga axial N foi encontrada
begin
fMtempx:=0;
fMtempy:=0;

```

```

fMtempX:=MomentoXXConcreto(CoordSec,ffcd,fxalfa,fxEpsilonC2,fNTensaoConcret
o);
//Primeiro, calculando Mxx

fMtempY:=MomentoYYConcreto(CoordSec,ffcd,fxalfa,fxEpsilonC2,fNTensaoConcret
o);
//Entao, calculando Myy
if fbArmPas then
begin
for fTemp:=0 to Length(CoordArmPas)-1 do // Inicio
momento do aço
begin
ahsi[fTemp]:=fvmax-CoordArmPas[fTemp].yi;
aEpsilonpsi[fTemp]:=fEpsilon1-
ahsi[fTemp]*(fEpsilon1-fEpsilon2)/fhalfa;

fvarMtemp:=aAs[fTemp]*TensaoArmPas(aEpsilonpsi[fTemp],fEs,ffyd); // Calcula-
se a força na barra e ela é salva numa variavel temporaria

fMtempX:=fMtempX+fvarMtemp*CoordArmPas[fTemp].yi;

fMtempY:=fMtempY+fvarMtemp*CoordArmPas[fTemp].xi;
end; //fim do momento do aço
end;
if fbArmAtiv then
begin
for fTemp:=0 to Length(CoordArmAtivT)-1 do //
Inicio força da armadura ativa
begin
ahpi[fTemp]:=fvmax-CoordArmAtiv[fTemp].yi;
aEpsilonpsi[fTemp]:=fEpsilon1-
ahpi[fTemp]*(fEpsilon1-fEpsilon2)/fhalfa;

varTemp:=aEpsilonpsi[fTemp]+aEpsilonprealong[fTemp];

fvarMtemp:=aAp[fTemp]*TensaoArmAtiv(fEpsilonpyd,fEpsilonpu,ffpyd,ffptd,varT
emp); // Calcula-se a força na barra e ela é salva numa variavel temporaria

fMtempX:=fMtempX+fvarMtemp*CoordArmAtiv[fTemp].yi;

fMtempY:=fMtempY+fvarMtemp*CoordArmAtiv[fTemp].xi;
end; //fim força da armadura ativa
end;

if abs(fMtempX)<0.000000001 then fMtempX:=0;
if abs(fMtempY)<0.000000001 then fMtempY:=0;

fMx:=fMtempX*cos(-
falfatemp*3.14159265359/180)+fMtempY*sin(-falfatemp*3.14159265359/180);
//Girando de U para X
fMy:=-fMtempX*sin(-
falfatemp*3.14159265359/180)+fMtempY*cos(-falfatemp*3.14159265359/180);
//Girando de V para Y

if fModoDeRuptura=0 then
MemoELU.Append('Angulo='+floattostr(falfatemp)+';
Eps1='+floattostrf(fEpsilon1,ffGeneral,5,4)+';
Eps2='+floattostrf(fEpsilon2,ffGeneral,5,4)+';
x/h='+floattostrf(fxalfa/fhalfa,ffGeneral,5,4)+';
força='+FloattostrF(fpNNow,ffGeneral,5,4)+'; Mxx =

```

```

'+FloattostrF(fMx,ffGeneral,5,4)+'; Myy =
'+FloattostrF(fMy,ffGeneral,5,4));
        if fModoDeRuptura=1 then
MemoELU.Append('Angulo='+floattostr(falfatemp)+';
Eps1='+floattostrf(fEpsilon1,ffGeneral,5,4)+';
Eps2='+floattostrf(fEpsilon2,ffGeneral,5,4)+';
x/d='+floattostrf(fxalfa/fdalfa,ffGeneral,5,4)+';
força='+FloattostrF(fpNNow,ffGeneral,5,4)+'; Mxx =
'+FloattostrF(fMx,ffGeneral,5,4)+'; Myy =
'+FloattostrF(fMy,ffGeneral,5,4));

        Seriel.AddXY(fMx,fMy);

        end; // Fim do calculo dos momentos - se a carga axial N foi
encontrada

        end;

        falfatemp:=falfatemp+fELUalfa;
        ProgressBarELU.Position:=trunc(falfatemp*100/360);
until falfatemp>360;
SeriePonto.AddXY(StrToFloat(EdtELUMxx.Text),StrToFloat(EdtELUMyy.Text));
LabelStatusELU.Caption:='Status: Ocioso';
LabelStatusELU.Refresh;

end;
end;

```

## B.2 ROTINA DA RELAÇÃO MOMENTO-CURVATURA

Esta rotina tem por objetivo obter uma série de valores de curvatura e seu respectivo momento resistente. Apresenta-se a rotina elaborada neste trabalho:

```

procedure TFrmMainFrm.ButGoMomCurvClick(Sender: TObject);
var
iContador:integer;
fpEpsilonNow,fpEpsilonPrevious,fpNNow,fpNPrevious,fpNAuxiliar,fpEpsilonAuxiliar
,fpdEpsilC2,fpNmax,fpNmin:real;

begin

if fVarBin=True then
begin
LabelStatusELS.Caption:='Status: Calculando... Aguarde';
LabelStatusELS.Refresh;
RotacaoDePontos(falfatemp,CoordSecT,CoordSec);
if fbArmPas then
RotacaoDePontos(falfatemp,CoordArmPasT,CoordArmPas);
if fbArmAtiv then
RotacaoDePontos(falfatemp,CoordArmAtivT,CoordArmAtiv);
PontosMaximoMinimo(CoordSec,fvmax,fvmin);
fhalfa:=fvmax-fvmin;
if fbArmPas then PontosMaximoMinimo(CoordArmPas,fAvmax,fAvmin);
if fbArmPas then fdalfa:=fAvmax-fAvmin;

// Inicio da preparação para iteração

```

```

Serie1.Clear;
MemoELS.Clear;
LabelChartX.Caption:='1/r (10/m)';
LabelChartYSup.Caption:='Md';
LabelChartYinf.Caption:='(kN.cm)';
ProgressBarELS.Position:=0;
fVarredura:=1;
fDeltaEps:=0;

repeat
  fCurvatura:=fDeltaEps/fhalfa/10; // = 1/r - unidade = 1/m - cuidado
ao manusear curvaturas
  fVarredura:=1;
  // N MINIMO
  fEpsilon2:=fEpsilonsu;
  fEpsilon1:=fEpsilon2+fDeltaEps;

  if fEpsilon1=fEpsilon2 then
    begin
      fxEpsilonC2:=10e6;
      fxalfa:=fxEpsilonC2*fEpsilon1/fEpsilonC2;
    end;
  if fEpsilon1<>fEpsilon2 then
    begin
      fxalfa:=fhalfa*fEpsilon1/(fEpsilon1-fEpsilon2);
      fxEpsilonC2:=fhalfa*fEpsilonC2/(fEpsilon1-fEpsilon2);
//Explicacoes na secao 4.1.2
    end;

fpNmin:=ForcaConcreto(CoordSec, ffcd, fxalfa, fxEpsilonC2, fNTensaoConcreto);

  if fbArmPas then
    begin
      for fTemp:=0 to Length(CoordArmPasT)-1 do // Inicio força do
aço
        begin
          ahsi[fTemp]:=fvmax-CoordArmPas[fTemp].yi;
          aEpsilonsi[fTemp]:=fEpsilon1-ahsi[fTemp]*(fEpsilon1-
fEpsilon2)/fhalfa;
fpNmin:=fpNmin+aAs[fTemp]*TensaoArmPas(aEpsilonsi[fTemp], fEs, ffyd);
          end; //fim força do aço
        end;
      if fbArmAtiv then
        begin
          for fTemp:=0 to Length(CoordArmAtivT)-1 do // Inicio força do
aço
            begin
              ahpi[fTemp]:=fvmax-CoordArmAtiv[fTemp].yi;
              aEpsilonpsi[fTemp]:=fEpsilon1-ahpi[fTemp]*(fEpsilon1-
fEpsilon2)/fhalfa;
              varTemp:=aEpsilonpsi[fTemp]+aEpsilonprealong[fTemp];
fpNmin:=fpNmin+aAp[fTemp]*TensaoArmAtiv(fEpsilonpyd, fEpsilonpu, ffpyd, ffptd,
varTemp);
            end; //fim força do aço
          end;
        fpEpsilonPrevious:=fEpsilon2;
        fpNPrevious:=fpNmin;

```

```

// N MAXIMO
if fDeltaEps>=fEpsilonCU then
begin
    fEpsilon1:=fEpsilonCU;
    fEpsilon2:=fEpsilon1-fDeltaEps;
end;
if fDeltaEps<fEpsilonCU then
begin
    fpdEpsilC2:=(fEpsilonCU-fEpsilonC2)/fEpsilonCU*fhalfa;
//Pode-se excluir essa linha substituindo seu conteudo na linha seguinte
    fEpsilon1:=fEpsilonC2+fpdEpsilC2*fDeltaEps/fhalfa;
    fEpsilon2:=fEpsilon1-fDeltaEps;
end;

if fEpsilon1=fEpsilon2 then
begin
    fxEpsilonC2:=10e6;
    fxalfa:=fxEpsilonC2*fEpsilon1/fEpsilonC2;
end;
if fEpsilon1<>fEpsilon2 then
begin
    fxalfa:=fhalfa*fEpsilon1/(fEpsilon1-fEpsilon2);
    fxEpsilonC2:=fhalfa*fEpsilonC2/(fEpsilon1-fEpsilon2);
//Explicacoes na secao 4.1.2
end;

fpNmax:=ForcaConcreto(CoordSec,ffcd,fxalfa,fxEpsilonC2,fNTensaoConcreto);
if fbArmPas then
begin
    for fTemp:=0 to Length(CoordArmPasT)-1 do // Inicio força do
aço
        begin
            ahsi[fTemp]:=fvmax-CoordArmPas[fTemp].yi;
            aEpsilonpsi[fTemp]:=fEpsilon1-ahsi[fTemp]*(fEpsilon1-
fEpsilon2)/fhalfa;

fpNmax:=fpNmax+aAs[fTemp]*TensaoArmPas(aEpsilonpsi[fTemp],fEs,ffyd);
            end; //fim força do aço
        end;
    if fbArmAtiv then
        begin
            for fTemp:=0 to Length(CoordArmAtivT)-1 do // Inicio força do
aço
                begin
                    ahpi[fTemp]:=fvmax-CoordArmAtiv[fTemp].yi;
                    aEpsilonpsi[fTemp]:=fEpsilon1-ahpi[fTemp]*(fEpsilon1-
fEpsilon2)/fhalfa;
                    varTemp:=aEpsilonpsi[fTemp]+aEpsilonprealong[fTemp];

fpNmax:=fpNmax+aAp[fTemp]*TensaoArmAtiv(fEpsilonpyd,fEpsilonpu,ffpyd,ffptd,
varTemp);
                end; //fim força do aço
            end;
            fpEpsilonNow:=fEpsilon2;
            fpNNow:=fpNmax;
            fpEpsilonAuxiliar:=fpEpsilonNow;
            fpNAuxiliar:=fpNNow;

if fNdFixado>=fpNmax then
begin

```

```

        fVarredura:=3;
    end;

    if fNdFixado<=fpNmin then
        begin
            fVarredura:=3;
        end;

    while fVarredura=1 do
        begin
            fpEpsilonNow:=fpEpsilonNow- (fpEpsilonNow-
fpEpsilonPrevious) / (fpNNow-fpNPrevious) * (fpNNow-fNdFixado);
            if fpEpsilonNow>fEpsilonC2 then fpEpsilonNow:=fEpsilonC2;
            fpEpsilonPrevious:=fpEpsilonNow;
            fpNPrevious:=fpNNow;
            fEpsilon2:=fpEpsilonNow;
            fEpsilon1:=fEpsilon2+fDeltaEps;

            if abs(fpEpsilonNow-fpEpsilonPrevious)<=fEpsilonErroMomCurv
then //Erro
                begin
                    fVarredura:=2;
                end;

            if fEpsilon1=fEpsilon2 then
                begin
                    fxEpsilonC2:=10e6;
                    fxalfa:=fxEpsilonC2*fEpsilon1/fEpsilonC2;
                end;
            if fEpsilon1<>fEpsilon2 then
                begin
                    fxalfa:=fhalfa*fEpsilon1/(fEpsilon1-fEpsilon2);
                    fxEpsilonC2:=fhalfa*fEpsilonC2/(fEpsilon1-fEpsilon2);
//Explicacoes na secao 4.1.2
                end;

                // Arredondar os epsilon, pois eles estão adquirindo erros
(nas casas decimais avançadas
                fStrtemp:=FloattostrF(fEpsilon1,ffGeneral,8,6);
                fEpsilon1:=StrToFloat(fStrtemp);
                fStrtemp:=FloattostrF(fEpsilon2,ffGeneral,8,6);
                fEpsilon2:=StrToFloat(fStrtemp);
                if abs(fEpsilon1)<0.000000001 then fEpsilon1:=0;
                if abs(fEpsilon2)<0.000000001 then fEpsilon2:=0;
                if abs(fDeltaEps)<0.000000001 then fDeltaEps:=0;

                fpNNow:=0;

            fpNNow:=ForcaConcreto(CoordSec,ffcd,fxalfa,fxEpsilonC2,fNTensaoConcreto);
            if fbArmPas then
                begin
                    for fTemp:=0 to Length(CoordArmPasT)-1 do // Inicio
força do aço
                        begin
                            ahsi[fTemp]:=fvmax-CoordArmPas[fTemp].yi;
                            aEpsilonSi[fTemp]:=fEpsilon1-
ahsi[fTemp]*(fEpsilon1-fEpsilon2)/fhalfa;

            fpNNow:=fpNNow+aAs[fTemp]*TensaoArmPas(aEpsilonSi[fTemp],fEs,ffyd);
                            end; //fim força do aço
                        end;
                    end;
                end;
            end;

```

```

        if fbArmAtiv then
            begin
                for fTemp:=0 to Length(CoordArmAtivT)-1 do
                    begin
                        ahpi[fTemp]:=fvmax-CoordArmAtiv[fTemp].yi;
                        aEpsilonpsi[fTemp]:=fEpsilon1-
ahpi[fTemp]*(fEpsilon1-fEpsilon2)/fhalfa;

varTemp:=aEpsilonpsi[fTemp]+aEpsilonprealong[fTemp];

fpNNow:=fpNNow+aAp[fTemp]*TensaoArmAtiv(fEpsilonpyd,fEpsilonpu,ffpyd,ffptd,
varTemp);
                            end;
                        end;
                    if fpNPrevious<=fpNmin then // Este bloco evita que a forca
fique constante (no dominio 1, a regioao em que todas as armaduras estao
trancionadas, ha forca constante
                        begin
                            if fpNNow<=fpNmin then
                                begin
                                    fpNPrevious:=fpNAuxiliar;
                                    fpEpsilonPrevious:=fpEpsilonAuxiliar;
                                end;
                            end;
                        if fVarredura=2 then // Inicio do calculo dos momentos - se a
carga axial N foi encontrada
                            begin
                                fMtempx:=0;
                                fMtempy:=0;

                                //Primeiro, calculando Mxx

fMtempx:=MomentoXXConcreto(CoordSec,ffcd,fxalfa,fxEpsilonC2,fNTensaoConcret
o);

                                //Entao, calculando Myy

fMtempy:=MomentoYYConcreto(CoordSec,ffcd,fxalfa,fxEpsilonC2,fNTensaoConcret
o);

                                    if fDeltaEps=0 then // Essa condicao foi necessaria
para evitar um bug dentro do programa
                                        begin // Para entender o processo,
basta retirar essas 5 linhas e testar o programa
                                            fMtempx:=0; // Em algumas situacoes de
curvatura zero, ainda haveria momento resistido pelo concreto
                                            fMtempy:=0; // E isso nao pode ocorrer, pois
no centro geométrico o momento de inércia de primeira ordem vale zero.
                                        end;

                                    if fbArmPas then
                                        begin
                                            for fTemp:=0 to Length(CoordArmPas)-1 do // Inicio
momento do aço
                                                begin
                                                    ahsi[fTemp]:=fvmax-CoordArmPas[fTemp].yi;
                                                    aEpsilonpsi[fTemp]:=fEpsilon1-
ahsi[fTemp]*(fEpsilon1-fEpsilon2)/fhalfa;

```

```

fvarMtemp:=aAs[fTemp]*TensaoArmPas(aEpsilon1[fTemp],fEs,ffyd); // Calcula-
se a força na barra e ela é salva numa variavel temporaria

fMtempx:=fMtempx+fvarMtemp*CoordArmPas[fTemp].yi;

fMtempy:=fMtempy+fvarMtemp*CoordArmPas[fTemp].xi;
    end; //fim do momento do aço
end;
if fbArmAtiv then
begin
for fTemp:=0 to Length(CoordArmAtivT)-1 do //
Inicio força da armadura ativa
begin
    ahpi[fTemp]:=fvmax-
CoordArmAtiv[fTemp].yi;
    aEpsilonpsi[fTemp]:=fEpsilon1-
ahpi[fTemp]*(fEpsilon1-fEpsilon2)/fhalfa;

varTemp:=aEpsilonpsi[fTemp]+aEpsilonprealong[fTemp];

fvarMtemp:=aAp[fTemp]*TensaoArmAtiv(fEpsilonpyd,fEpsilonpu,ffpyd,ffptd,varT
emp); // Calcula-se a força na barra e ela é salva numa variavel temporaria

fMtempx:=fMtempx+fvarMtemp*CoordArmAtiv[fTemp].yi;

fMtempy:=fMtempy+fvarMtemp*CoordArmAtiv[fTemp].xi;
    end; //fim força da armadura ativa
end;

    fMx:=fMtempx*cos(-
falfatemp*3.14159265359/180)+fMtempy*sin(-falfatemp*3.14159265359/180);
//Girando de U para X
    fMy:=-fMtempx*sin(-
falfatemp*3.14159265359/180)+fMtempy*cos(-falfatemp*3.14159265359/180);
//Girando de V para Y

    if abs(fMx)<0.000000001 then fMtempx:=0;
    if abs(fMy)<0.000000001 then fMtempy:=0;
    if abs(fMy)>0.000000001 then MemoELS.Append('Existe
momento residual na direção perpendicular!');

MemoELS.Append('Eps1='+FloatToStrF(fEpsilon1,ffGeneral,6,5)+'';
Eps2='+FloatToStrF(fEpsilon2,ffGeneral,6,5)+'';
força='+FloatToStrF(fpNNow,ffGeneral,6,5)+'';
Curv.='+FloatToStrF(fCurvatura,ffGeneral,6,5)+''; Mxx =
'+FloatToStrF(fMtempx,ffGeneral,6,5)+''; Myy =
'+FloatToStrF(fMtempy,ffGeneral,6,5)');
    Serie1.AddXY(fCurvatura,fMx);

end; // Fim do calculo dos momentos - se a carga axial N
foi encontrada

end;
fDeltaEps:=fDeltaEps+0.1;

until fVarredura=3;

```

```

LabelStatusELS.Caption:='Status: Ocioso';
LabelStatusELS.Refresh;

end;
end;

```

### B.3 ROTINA DA VERIFICAÇÃO DE ESTADO-LIMITE DE SERVIÇO

O objetivo desta rotina é obter a conformação de deformações correspondentes a solicitação definida pelo usuário, como força axial e a dupla de momentos fletores  $M_{xx}$  e  $M_{yy}$ . Então, compara-se os resultados com os valores indicados pela norma ABNT NBR 6118:2014 para estados-limites de serviço.

```

procedure TFrmMainFrm.ButCalc2Click(Sender: TObject);
var
iContador:integer;
fpMxx,fpMyy,fpN,fpNNow,fpMxNow,fpMyNow,fpHx,fpHy,fpEpsilonC,fpDeltaEpsX,fpDeltaEpsY,fpEpsilonCPrevious,fpDeltaEpsXPrevious,fpDeltaEpsYPrevious,fpDeltaEpsAlfa,fpCurvatura,fpCurvaturaX,fpCurvaturaY,fpVarTemp1,fpVarTemp2:real;
aJacobn,aJacobInverso:array[1..3,1..3] of real;

begin
  if fVarBin=True then
    begin
      fpN:=strtofloat(EdtELSNd.Text);
      fpMxx:=strtofloat(EdtELSMxx.Text);
      fpMyy:=strtofloat(EdtELSMyy.Text);
      SetLength(CoordSec,PlanPolyg.RowCount); // Inicio da aquisição dos
dados - Dos Edit para as variaveis
      VetorReceberPontos(PlanPolyg,CoordSec);
      ObterCaractGeomet(CoordSec,fAc,fxcg,fycg); //Calculo das
caracteristicas geometricas
      if fbArmPas then
        begin
          SetLength(CoordArmPas,PlanArmPas.RowCount-1);
          SetLength(aAs,PlanArmPas.RowCount-1);
          SetLength(ahsi,PlanArmPas.RowCount-1);
          SetLength(aEpsilonsi,PlanArmPas.RowCount-1);
          VetorReceberPontosArmaduraPassiva(PlanArmPas,aAs,CoordArmPas);
// Fim da aquisição dos dados - Dos Edit para as variaveis
          SetLength(CoordArmPasT,Length(CoordArmPas));
          TranslacaoDePontos(CoordArmPas,fxcg,fycg,CoordArmPasT); //Fim
da translação dos pontos
          end;

        if fbArmAtiv then
          begin
            SetLength(CoordArmAtiv,PlanArmAtiv.RowCount-1);
            SetLength(aAp,PlanArmAtiv.RowCount-1);
            SetLength(ahpi,PlanArmAtiv.RowCount-1);
            SetLength(aEpsilonprealong,PlanArmAtiv.RowCount-1);
            SetLength(aEpsilonpsi,PlanArmAtiv.RowCount-1);
            SetLength(aDiametroAp,PlanArmAtiv.RowCount-1);

```

```

VetorReceberPontosArmaduraAtiva(PlanArmAtiv, aAp, aEpsilonprealong, aDiametroA
p, CoordArmAtiv); // Fim da aquisição dos dados - Dos Edit para as variaveis
    SetLength(CoordArmAtivT, Length(CoordArmAtiv));
    TranslacaoDePontos(CoordArmAtiv, fxcg, fycg, CoordArmAtivT);
//Fim da translação dos pontos
    end;
    SetLength(CoordSecT, Length(CoordSec)); //Inicio da translação dos
pontos
    TranslacaoDePontos(CoordSec, fxcg, fycg, CoordSecT);
    fAs:=0; //Inicio do calculo dos valores normais
    if fbArmPas then
    begin
        for fTemp:=1 to PlanArmPas.RowCount-1 do
        begin
            fAs:=fAs+Strtfloat(PlanArmPas.Cells[3, fTemp]);
        end;
    end;
    fSigmacd:=ffcd*0.85;
    fNdmax:=(fAc*fSigmacd+fAs*TensaoArmPas(fEpsilonC2, fEs, ffyd));
    fNdmin:=(-ffyd*fAs);

    if fbArmAtiv then
    begin
        for fTemp:=1 to PlanArmAtiv.RowCount-1 do
        begin
            varTemp:=(aEpsilonprealong[fTemp-1])+fEpsilonC2;
            fNdmax:=fNdmax+aAp[fTemp-
1]*TensaoArmAtiv(fEpsilonpyd, fEpsilonpu, ffpyd, ffptd, varTemp);
            varTemp:=(aEpsilonprealong[fTemp-1])-10
            fNdmin:=fNdmin+aAp[fTemp-
1]*TensaoArmAtiv(fEpsilonpyd, fEpsilonpu, ffpyd, ffptd, varTemp);
        end;
    end;
//    EdtNmax.Text:=FloatToStrF(fNdmax, ffGeneral, 8, 6);
//    EdtNmin.Text:=FloatttostrF(fNdmin, ffGeneral, 8, 6); //Fim do
calculo dos valores normais
    if fpN>=fNdmax then
    begin
        LabelStatusProcurar2.Caption:='Ruptura por compressão atingida.
Força de compressão máxima='+FloatToStrF(fNdmax, ffGeneral, 8, 6);
        fVarBin:=False;
    end;
    if fpN<=fNdmin then
    begin
        LabelStatusProcurar2.Caption:='Ruptura por tração atingida.
Força de tração máxima='+FloatToStrF(fNdmin, ffGeneral, 8, 6);
        fVarBin:=False;
    end;
end;

    if fVarBin=True then
    begin
        MemoProcurar2.Clear;
        RotacaoDePontos(0, CoordSecT, CoordSec);
        if fbArmPas then
        RotacaoDePontos(0, CoordArmPasT, CoordArmPas);
        if fbArmAtiv then
        RotacaoDePontos(0, CoordArmAtivT, CoordArmAtiv);
        PontosMaximoMinimo(CoordSec, fpVarTemp1, fpVarTemp2);
    end;

```

```

fphy:=fpVarTemp1-fpVarTemp2;

RotacaoDePontos(90,CoordSecT,CoordSec);
if fbArmPas then
RotacaoDePontos(90,CoordArmPasT,CoordArmPas);
if fbArmAtiv then
RotacaoDePontos(90,CoordArmAtivT,CoordArmAtiv);
PontosMaximoMinimo(CoordSec,fpVarTemp1,fpVarTemp2);
fphx:=fpVarTemp1-fpVarTemp2;

//Encontrou-se fphy e fphx, agora, calcula-se o
jacobiano.
for iContador:=1 to 4 do
begin
//Para isso, calculam-se 4 estados de deformacao:
// 1 - falfatemp=0; Epsilon1 = 0.1; Epsilon2 =
0.1
// 2 - falfatemp=0; Epsilon1 = 0.2; Epsilon2 =
0.2 (isto e, varia-se o fpEpsilonC em 0.1)
// 3 - falfatemp=90; Epsilon1 = 0.2; Epsilon2 =
0.0 (isto e, varia-se o fpCurvaturaY em 0.2)
// 4 - falfatemp=0; Epsilon1 = 0.2; Epsilon2 =
0.0 (isto e, varia-se o fpCurvaturaX em 0.2)

if iContador=1 then
begin
falfatemp:=0;
fEpsilon1:=0.1;
fEpsilon2:=0.1;
end;
if iContador=2 then
begin
falfatemp:=0;
fEpsilon1:=0.2;
fEpsilon2:=0.2;
end;
if iContador=3 then
begin
falfatemp:=90;
fEpsilon1:=0.2;
fEpsilon2:=0.0;
end;
if iContador=4 then
begin
falfatemp:=0;
fEpsilon1:=0.2;
fEpsilon2:=0.0;
end;

RotacaoDePontos(falfatemp,CoordSecT,CoordSec);
//inicio, obtencao de N, Mxx e Myy a partir de falfatemp, fEpsilon1 e
fEpsilon2;
if fbArmPas then
RotacaoDePontos(falfatemp,CoordArmPasT,CoordArmPas);
if fbArmAtiv then
RotacaoDePontos(falfatemp,CoordArmAtivT,CoordArmAtiv);
PontosMaximoMinimo(CoordSec,fvmax,fvmin);
fhalfa:=fvmax-fvmin;
if abs(fEpsilon1)<0.000000001 then fEpsilon1:=0;
if abs(fEpsilon2)<0.000000001 then fEpsilon2:=0;
if fEpsilon1=fEpsilon2 then

```

```

begin
    fxEpsilonC2:=10e6;
    fxalfa:=fxEpsilonC2*fEpsilon1/fEpsilonC2;
end;
if fEpsilon1<>fEpsilon2 then
begin
    fxalfa:=fhalfa*fEpsilon1/(fEpsilon1-
fEpsilon2);

fxEpsilonC2:=fhalfa*fEpsilonC2/(fEpsilon1-fEpsilon2); //Explicacoes na
secao 4.1.2

end;

fNdtemp:=ForcaConcreto(CoordSec, ffcd, fxalfa, fxEpsilonC2, fNTensaoConcreto);

fMtempX:=MomentoXXConcreto(CoordSec, ffcd, fxalfa, fxEpsilonC2, fNTensaoConcret
o);

fMtempY:=MomentoYYConcreto(CoordSec, ffcd, fxalfa, fxEpsilonC2, fNTensaoConcret
o);

if fbArmPas then
begin
    for fTemp:=0 to Length(CoordArmPasT)-1 do
// Inicio força do aço
begin
    CoordArmPas[fTemp].yi;
    ahsi[fTemp]:=fvmax-
    aEpsilonSi[fTemp]:=fEpsilon1-
    ahsi[fTemp]*(fEpsilon1-fEpsilon2)/fhalfa;

    fvarMtemp:=aAs[fTemp]*TensaoArmPas(aEpsilonSi[fTemp], fEs, ffyd); // Calcula-
se a força na barra e ela é salva numa variavel temporaria
    fNdtemp:=fNdtemp+fvarMtemp;

    fMtempX:=fMtempX+fvarMtemp*CoordArmPas[fTemp].yi;

    fMtempY:=fMtempY+fvarMtemp*CoordArmPas[fTemp].xi;
end; //fim força do aço
end;

if fbArmAtiv then
begin
    for fTemp:=0 to Length(CoordArmAtivT)-1 do
// Inicio força do aço
begin
    CoordArmAtiv[fTemp].yi;
    ahpi[fTemp]:=fvmax-
    aEpsilonPsi[fTemp]:=fEpsilon1-
    ahpi[fTemp]*(fEpsilon1-fEpsilon2)/fhalfa;

    varTemp:=aEpsilonPsi[fTemp]+aEpsilonPrealong[fTemp];

    fvarMtemp:=aAp[fTemp]*TensaoArmAtiv(fEpsilonPyd, fEpsilonPu, ffpyd, ffptd, varT
emp); // Calcula-se a força na barra e ela é salva numa variavel temporaria
    fNdtemp:=fNdtemp+fvarMtemp;

    fMtempX:=fMtempX+fvarMtemp*CoordArmAtiv[fTemp].yi;

    fMtempY:=fMtempY+fvarMtemp*CoordArmAtiv[fTemp].xi;
end; //fim força do aço

```

```

end;

if abs(fMtempx)<0.000001 then fMtempx:=0;
if abs(fMtempy)<0.000001 then fMtempy:=0;

fMx:=fMtempx*cos(-
falfatemp*3.14159265359/180)+fMtempy*sin(-falfatemp*3.14159265359/180);
//Girando de U para X
fMy:=-fMtempx*sin(-
falfatemp*3.14159265359/180)+fMtempy*cos(-falfatemp*3.14159265359/180);
//Girando de V para Y

if abs(fMx)<0.000001 then fMx:=0;
if abs(fMy)<0.000001 then fMy:=0;

if iContador=1 then
begin
aJacobn[1,1]:=-fNdtemp;
aJacobn[1,2]:=-fMx;
aJacobn[1,3]:=-fMy;
aJacobn[2,1]:=-fNdtemp;
aJacobn[2,2]:=-fMx;
aJacobn[2,3]:=-fMy;
aJacobn[3,1]:=-fNdtemp;
aJacobn[3,2]:=-fMx;
aJacobn[3,3]:=-fMy;
fpNNow:=fNdtemp; // Guardando esses 3
valores, para serem utilizados novamente, ver iContador=4
fpMxNow:=fMx;
fpMyNow:=fMy;
end;
if iContador=2 then
begin
aJacobn[1,1]:=(fNdtemp+aJacobn[1,1])/0.1;
aJacobn[1,2]:=(fMx+aJacobn[1,2])/0.1;;
aJacobn[1,3]:=(fMy+aJacobn[1,3])/0.1;;
end;
if iContador=3 then
begin
aJacobn[2,1]:=(fNdtemp+aJacobn[2,1])/0.2;
aJacobn[2,2]:=(fMx+aJacobn[2,2])/0.2;
aJacobn[2,3]:=(fMy+aJacobn[2,3])/0.2;
end;
if iContador=4 then
begin
aJacobn[3,1]:=(fNdtemp+aJacobn[3,1])/0.2;
aJacobn[3,2]:=(fMx+aJacobn[3,2])/0.2;
aJacobn[3,3]:=(fMy+aJacobn[3,3])/0.2;
// Para começar a iteracao, preciso
comparar os valores iniciais, que foram obtidos no iContador=1, repare que
salvei-os nas variaveis fpNNow, fpMxNow e fpMyNow, vou agora devolve-las
para liberar as variaveis fNdtemp e as suas analogas para momento, os
valores foram salvos no iContador=1
fNdtemp:=fpNNow;
fMx:=fpMxNow;
fMy:=fpMyNow;
fpEpsilonC:=0.1;
fpDeltaEpsX:=0.0;
fpDeltaEpsY:=0.0;
end;
end;

```

```

// Agora, pode-se fazer o processo iterativo
// O pascal nao faz divisao de matrizes, entao o
Jacobiano deve ser invertido antes
// O jacobiano invertido sera mantido na variavel
Jacoboinv
varTemp:=aJacobn[1,1]*(aJacobn[2,2]*aJacobn[3,3]-
aJacobn[2,3]*aJacobn[3,2])-aJacobn[1,2]*(aJacobn[3,3]*aJacobn[2,1]-
aJacobn[2,3]*aJacobn[3,1])+aJacobn[1,3]*(aJacobn[2,1]*aJacobn[3,2]-
aJacobn[2,2]*aJacobn[3,1]);
aJacobInverso[1,1]:=(aJacobn[2,2]*aJacobn[3,3]-
aJacobn[2,3]*aJacobn[3,2])/varTemp;
aJacobInverso[2,1]:=- (aJacobn[2,1]*aJacobn[3,3]-
aJacobn[2,3]*aJacobn[3,1])/varTemp;
aJacobInverso[3,1]:=(aJacobn[2,1]*aJacobn[3,2]-
aJacobn[2,2]*aJacobn[3,1])/varTemp;
aJacobInverso[1,2]:=- (aJacobn[1,2]*aJacobn[3,3]-
aJacobn[1,3]*aJacobn[3,2])/varTemp;
aJacobInverso[2,2]:=(aJacobn[1,1]*aJacobn[3,3]-
aJacobn[1,3]*aJacobn[3,1])/varTemp;
aJacobInverso[3,2]:=- (aJacobn[1,1]*aJacobn[3,2]-
aJacobn[1,2]*aJacobn[3,1])/varTemp;
aJacobInverso[1,3]:=(aJacobn[1,2]*aJacobn[2,3]-
aJacobn[1,3]*aJacobn[2,2])/varTemp;
aJacobInverso[2,3]:=- (aJacobn[1,1]*aJacobn[2,3]-
aJacobn[1,3]*aJacobn[2,1])/varTemp;
aJacobInverso[3,3]:=(aJacobn[1,1]*aJacobn[2,2]-
aJacobn[1,2]*aJacobn[2,1])/varTemp;
fpEpsilonCPrevious:=0.1;
fpDeltaEpsXPrevious:=0.2;
fpDeltaEpsYPrevious:=0.2;
iContador:=0; // Agora o iContador ira contar o numero de
iteracoes - caso exceder 20000 a iteracao ira parar
while
abs(fpEpsilonCPrevious+fpDeltaEpsXPrevious+fpDeltaEpsYPrevious)>=fEpsilonEr
roELS do
begin
fpNNow:=fNdtemp-fpN;
fpMxNow:=fMx-fpMxx;
fpMyNow:=fMy-fpMyy;
iContador:=iContador+1;
//Agora, a divisao de matrizes (produto do
inverso, na verdade)

fpEpsilonCPrevious:=(aJacobInverso[1,1]*fpNNow+aJacobInverso[1,2]*fpMxNow+a
JacobInverso[1,3]*fpMyNow)/2;
fpEpsilonC:=fpEpsilonC-fpEpsilonCPrevious;

fpDeltaEpsXPrevious:=(aJacobInverso[2,1]*fpNNow+aJacobInverso[2,2]*fpMxNow+
aJacobInverso[2,3]*fpMyNow)/2;
fpDeltaEpsX:=fpDeltaEpsX-fpDeltaEpsXPrevious;

fpDeltaEpsYPrevious:=(aJacobInverso[3,1]*fpNNow+aJacobInverso[3,2]*fpMxNow+
aJacobInverso[3,3]*fpMyNow)/2;
fpDeltaEpsY:=fpDeltaEpsY-fpDeltaEpsYPrevious;

fpCurvaturaX:=fpDeltaEpsX/fphx;
fpCurvaturaY:=fpDeltaEpsY/fphy;
if fpCurvaturaY<0 then
begin
if fpCurvaturaX>0 then
falfatemp:=arctan(fpCurvaturaX/fpCurvaturaY)*180/pi+180;

```

```

        if fpCurvaturaX<0 then
falfatemp:=arctan(fpCurvaturaX/fpCurvaturaY)*180/pi+180;
        if fpCurvaturaX=0 then falfatemp:=180;
        end;
        if fpCurvaturaY>0 then
        begin
            if fpCurvaturaX>0 then
falfatemp:=arctan(fpCurvaturaX/fpCurvaturaY)*180/pi;
            if fpCurvaturaX<0 then
falfatemp:=arctan(fpCurvaturaX/fpCurvaturaY)*180/pi+360;
            if abs(fpCurvaturaX)<=0.00001 then
falfatemp:=0;
        end;
        if abs(fpCurvaturaY)<=0.00001 then
        begin
            if fpCurvaturaX>0 then falfatemp:=90;
            if fpCurvaturaX<0 then falfatemp:=270;
            if abs(fpCurvaturaX)<=0.00001 then
falfatemp:=0;
        end;

fpCurvatura:=sqrt(fpCurvaturaX*fpCurvaturaX+fpCurvaturaY*fpCurvaturaY);
        RotacaoDePontos(falfatemp,CoordSecT,CoordSec);
        PontosMaximoMinimo(CoordSec,fvmax,fvmin);
        fhalfa:=fvmax-fvmin;
        fpDeltaEpsAlfa:=fpCurvatura*fhalfa;
        fEpsilon1:=fpEpsilonC+fpDeltaEpsAlfa/2;
        fEpsilon2:=fpEpsilonC-fpDeltaEpsAlfa/2;

        //Obtencao N e Mx e My
        RotacaoDePontos(falfatemp,CoordSecT,CoordSec);
//inicio, obtencao de N, Mxx e Myy a partir de falfatemp, fEpsilon1 e
fEpsilon2;
        if fbArmPas then
RotacaoDePontos(falfatemp,CoordArmPasT,CoordArmPas);
        if fbArmAtiv then
RotacaoDePontos(falfatemp,CoordArmAtivT,CoordArmAtiv);
        PontosMaximoMinimo(CoordSec,fvmax,fvmin);
        fhalfa:=fvmax-fvmin;
        if abs(fEpsilon1)<0.000000001 then fEpsilon1:=0;
        if abs(fEpsilon2)<0.000000001 then fEpsilon2:=0;
        if fEpsilon1=fEpsilon2 then
        begin
            fxEpsilonC2:=10e6;
            fxalfa:=fxEpsilonC2*fEpsilon1/fEpsilonC2;
        end;
        if fEpsilon1<>fEpsilon2 then
        begin
            fxalfa:=fhalfa*fEpsilon1/(fEpsilon1-
fEpsilon2);
            fxEpsilonC2:=fhalfa*fEpsilonC2/(fEpsilon1-
fEpsilon2); //Explicacoes na secao 4.1.2
        end;

fNdtemp:=ForcaConcreto(CoordSec,ffcd,fxalfa,fxEpsilonC2,fNTensaoConcreto);

fMtempx:=MomentoXXConcreto(CoordSec,ffcd,fxalfa,fxEpsilonC2,fNTensaoConcret
o);

```

```

fMtempy:=MomentoYYConcreto(CoordSec,ffcd,fxalfa,fxEpsilonC2,fNTensaoConcret
o);

        if fbArmPas then
            begin
                for fTemp:=0 to Length(CoordArmPasT)-1 do
// Inicio força do aço
                    begin
                        ahsi[fTemp]:=fvmax-
CoordArmPas[fTemp].yi;
                        aEpsilon1[fTemp]:=fEpsilon1-
ahsi[fTemp]*(fEpsilon1-fEpsilon2)/fhalfa;

fvarMtemp:=aAs[fTemp]*TensaoArmPas(aEpsilon1[fTemp],fEs,ffyd); // Calcula-
se a força na barra e ela é salva numa variavel temporaria
                        fNdtemp:=fNdtemp+fvarMtemp;

fMtempx:=fMtempx+fvarMtemp*CoordArmPas[fTemp].yi;

fMtempy:=fMtempy+fvarMtemp*CoordArmPas[fTemp].xi;
                        end; //fim força do aço
                    end;

            if fbArmAtiv then
                begin
                    for fTemp:=0 to Length(CoordArmAtivT)-1 do
// Inicio força do aço
                        begin
                            ahpi[fTemp]:=fvmax-
CoordArmAtiv[fTemp].yi;
                            aEpsilonpsi[fTemp]:=fEpsilon1-
ahpi[fTemp]*(fEpsilon1-fEpsilon2)/fhalfa;

varTemp:=aEpsilonpsi[fTemp]+aEpsilonprealong[fTemp];

fvarMtemp:=aAp[fTemp]*TensaoArmAtiv(fEpsilonpyd,fEpsilonpu,ffpyd,ffptd,varT
emp); // Calcula-se a força na barra e ela é salva numa variavel temporaria
                            fNdtemp:=fNdtemp+fvarMtemp;

fMtempx:=fMtempx+fvarMtemp*CoordArmAtiv[fTemp].yi;

fMtempy:=fMtempy+fvarMtemp*CoordArmAtiv[fTemp].xi;
                            end; //fim força do aço
                        end;

                    if abs(fMtempx)<0.000001 then fMtempx:=0;
                    if abs(fMtempy)<0.000001 then fMtempy:=0;

                    fMx:=fMtempx*cos(-
falfatemp*3.14159265359/180)+fMtempy*sin(-falfatemp*3.14159265359/180);
//Girando de U para X
                    fMy:=-fMtempx*sin(-
falfatemp*3.14159265359/180)+fMtempy*cos(-falfatemp*3.14159265359/180);
//Girando de V para Y

                    if abs(fMx)<0.000001 then fMx:=0;
                    if abs(fMy)<0.000001 then fMy:=0;

                    //Fim da obtencao N Mx e My
                    if iContador=20000 then

```

```

begin
    fpEpsilonCPrevious:=0;
    fpDeltaEpsXPrevious:=0;
    fpDeltaEpsYPrevious:=0;
end;

end;

end;
//while acaba aqui
fCurvatura:=(fEpsilon1-fEpsilon2)/fhalfa/10;
MemoProcurar2.Append('Eps1 =
'+FloatToStrF(fEpsilon1,ffGeneral,5,4)+'; Eps2 =
'+FloatToStrF(fEpsilon2,ffGeneral,5,4)+'; Angulo =
'+FloatToStrF(falfatemp,ffGeneral,5,4)+'; Curv 1/r (1/m) =
'+FloattostrF(fCurvatura,ffGeneral,6,4));
MemoProcurar2.Append('Numero de iteracoes =
'+FloatToStr(iContador));
MemoProcurar2.Append('');
MemoProcurar2.Append('ESTADO-LIMITE DE SERVIÇO: ABERTURA
DE FISSURAS');
MemoProcurar2.Append('');

//Agora vou incluir as tensões e abertura de fissura em
cada armadura
if fbArmPas then
begin
MemoProcurar2.Append(' Abertura de fissuras na
armadura passiva:');
for fTemp:=0 to Length(CoordArmPas)-1 do // Inicio
momento do aço
begin
ahsi[fTemp]:=fvmax-CoordArmPas[fTemp].yi;
aEpsilonsi[fTemp]:=fEpsilon1-
ahsi[fTemp]*(fEpsilon1-fEpsilon2)/fhalfa;
fvarMtemp:=0;
if aAs[fTemp]>0 then
fvarMtemp:=exp(ln(aAs[fTemp]*4/3.14159265359)*0.5)*10; //Bitola (mm) -
calculada a partir da area
fvarMtemp:=fvarMtemp/(12.5*2.25); //Primeiro
termo

fvarMtemp:=fvarMtemp*TensaoArmPas(aEpsilonsi[fTemp],fEs,ffyd)/fEs; //
Segundo termo

fvarMtemp:=fvarMtemp*3*TensaoArmPas(aEpsilonsi[fTemp],fEs,ffyd)/(0.1*0.3*ex
p(ln(ffck)*0.666667)); // Terceiro termo
if aEpsilonsi[fTemp]>0 then fvarMtemp:=0;
MemoProcurar2.Append(' Barra nr.:
'+FloatToStrF(fTemp+1,ffGeneral,6,4)+'; Tensao (KN/cm²):
'+floattostrF(TensaoArmPas(aEpsilonsi[fTemp],fEs,ffyd),ffGeneral,6,4)+';
wk,1 (mm) : '+floattostrF(fvarMtemp,ffGeneral,6,4));
end; //fim da abertura de fissuras da armadura
passiva

end;
if fbArmPas=false then
begin
MemoProcurar2.Append(' Nao ha armadura passiva para
calculo da abertura de fissura');
end;
end;

```

```

        if fbArmAtiv then
            begin
                MemoProcurar2.Append('');
                MemoProcurar2.Append('  Abertura de fissuras na
armadura ativa:');
                for fTemp:=0 to Length(CoordArmAtiv)-1 do
                    begin
                        ahpi[fTemp]:=fvmax-CoordArmAtiv[fTemp].yi;
                        aEpsilonpsi[fTemp]:=fEpsilon1-
ahpi[fTemp]*(fEpsilon1-fEpsilon2)/fhalfa;

varTemp:=aEpsilonpsi[fTemp]+aEpsilonprealong[fTemp];
                        fvarMtemp:=0;
                        fvarMtemp:=aDiametroAp[fTemp]/(12.5*fEtaP1);
//Primeiro termo - Bitola (mm) - deve ser fornecida pelo usuario

fvarMtemp:=fvarMtemp*(TensaoArmAtiv(fEpsilonpyd,fEpsilonpu,ffpyd,ffptd,varT
emp)-
TensaoArmAtiv(fEpsilonpyd,fEpsilonpu,ffpyd,ffptd,aEpsilonprealong[fTemp]))/
fEp; // Segundo termo

fvarMtemp:=fvarMtemp*3*(TensaoArmAtiv(fEpsilonpyd,fEpsilonpu,ffpyd,ffptd,va
rTemp)-
TensaoArmAtiv(fEpsilonpyd,fEpsilonpu,ffpyd,ffptd,aEpsilonprealong[fTemp]))/
(0.1*0.3*exp(ln(ffck)*0.666667)); // Terceiro termo
                        if aEpsilonpsi[fTemp]>0 then fvarMtemp:=0;
                        MemoProcurar2.Append('  Barra nr.:
'+FloatToStrF(fTemp+1,ffGeneral,6,4)+'; Tensao (KN/cm²):
'+floattostrF((TensaoArmAtiv(fEpsilonpyd,fEpsilonpu,ffpyd,ffptd,varTemp)-
TensaoArmAtiv(fEpsilonpyd,fEpsilonpu,ffpyd,ffptd,aEpsilonprealong[fTemp])),
ffGeneral,6,4)+'; wk,1 (mm) : '+floattostrF(fvarMtemp,ffGeneral,6,4));
                        end; //fim da abertura de fissuras da armadura
passiva

                    end;
                if fbArmAtiv=false then
                    begin
                        MemoProcurar2.Append('  Nao ha armadura ativa para
calculo da abertura de fissura');
                    end;

                MemoProcurar2.Append('');
                MemoProcurar2.Append('');
                MemoProcurar2.Append('ESTADO-LIMITE DE FORMACAO DE
FISSURAS');
                MemoProcurar2.Append('Menor deformacao que ocorre na secao
= '+FloatToStrF(fEpsilon2,ffGeneral,5,4));
                MemoProcurar2.Append('Menor deformacao permitida =
'+FloatToStrF(-fEpsilonELSF,ffGeneral,5,4));
                if fEpsilon2>=-fEpsilonELSF then
MemoProcurar2.Append('ELS-F: ATENDIDA. ');
                if fEpsilon2<-fEpsilonELSF then MemoProcurar2.Append('ELS-
F: NAO ATENDIDA. ');
                MemoProcurar2.Append('');
                MemoProcurar2.Append('');
                MemoProcurar2.Append('ESTADO-LIMITE DE DESCOMPRESSAO');
                MemoProcurar2.Append('Menor deformacao que ocorre na secao
= '+FloatToStrF(fEpsilon2,ffGeneral,5,4));
                MemoProcurar2.Append('Menor deformacao permitida = 0');

```

```

        if fEpsilon2>=0 then MemoProcurar2.Append('ELS-D:
ATENDIDA.');
```

```

        if fEpsilon2<0 then MemoProcurar2.Append('ELS-D: NAO
ATENDIDA.');
```

```

        MemoProcurar2.Append(' ');
        if iContador=20000 then MemoProcurar2.Clear;
        if iContador=20000 then MemoProcurar2.Append('Processo
travou - resultados desconhecidos');
```

```

        if iContador=20000 then MemoProcurar2.Append('As
solicitacoes podem nao atender o ELU!');
```

```

end;
```

```

end.
```

#### B.4 ROTINA PARA CÁLCULAR SOLICITAÇÕES

Esta rotina calcula as solicitações correspondentes à uma certa conformação de deformações.

```

procedure TFrmMainFrm.ButProcurarClick(Sender: TObject);
var
iContador:integer;

begin
  fVarBin:=True; // Inicio das medidas defensivas

  if fVarBin=True then
    begin
      SetLength(CoordSec,PlanPolyg.RowCount); // Inicio da aquisição dos
dados - Dos Edit para as variaveis
      VetorReceberPontos(PlanPolyg,CoordSec);
      ObterCaractGeomet(CoordSec,fAc,fxcg,fycg); //Calculo das
caracteristicas geometricas
      SetLength(CoordSecT,Length(CoordSec)); //Inicio da translação dos
pontos
      TranslacaoDePontos(CoordSec,fxcg,fycg,CoordSecT);
      if fbArmPas then
        begin
          SetLength(CoordArmPas,PlanArmPas.RowCount-1);
          SetLength(aAs,PlanArmPas.RowCount-1);
          SetLength(ahsi,PlanArmPas.RowCount-1);
          SetLength(aEpsilonSi,PlanArmPas.RowCount-1);
          VetorReceberPontosArmaduraPassiva(PlanArmPas,aAs,CoordArmPas);
// Fim da aquisição dos dados - Dos Edit para as variaveis
          SetLength(CoordArmPasT,Length(CoordArmPas));
          TranslacaoDePontos(CoordArmPas,fxcg,fycg,CoordArmPasT); //Fim
da translação dos pontos
          end;

        if fbArmAtiv then
          begin
            SetLength(CoordArmAtiv,PlanArmAtiv.RowCount-1);
            SetLength(aAp,PlanArmAtiv.RowCount-1);
            SetLength(ahpi,PlanArmAtiv.RowCount-1);
            SetLength(aEpsilonprealong,PlanArmAtiv.RowCount-1);
            SetLength(aEpsilonpsi,PlanArmAtiv.RowCount-1);
            SetLength(aDiametroAp,PlanArmAtiv.RowCount-1);
```

```

VetorReceberPontosArmaduraAtiva(PlanArmAtiv, aAp, aEpsilonprealong, aDiametroA
p, CoordArmAtiv); // Fim da aquisição dos dados - Dos Edit para as variaveis
    SetLength(CoordArmAtivT, Length(CoordArmAtiv));
    TranslacaoDePontos(CoordArmAtiv, fxcg, fycg, CoordArmAtivT);
//Fim da translação dos pontos
end;

fEpsilon1:=StrToFloat(EdtProcEps1.Text); // Este bloco inverte
Epsilon1 e Epsilon2 automaticamente caso Eps2 seja maior que Eps1
fEpsilon2:=StrToFloat(EdtProcEps2.Text);
if fEpsilon2>fEpsilon1 then begin
    falfatemp:=StrToFloat(EdtProcAlfa.Text);
    falfatemp:=falfatemp+180;
    EdtProcAlfa.Text:=Floattostr(falfatemp);
    EdtProcEps1.Text:=FloatToStr(fEpsilon2);
    EdtProcEps2.Text:=FloatToStr(fEpsilon1);
end;

if fVarBin=True then begin
    // Inicio da preparação para iteração
    falfatemp:=StrToFloat(EdtProcAlfa.Text);
    fEpsilon1:=StrToFloat(EdtProcEps1.Text);
    fEpsilon2:=StrToFloat(EdtProcEps2.Text);
    RotacaoDePontos(falfatemp, CoordSecT, CoordSec);
    if fbArmPas then
        RotacaoDePontos(falfatemp, CoordArmPasT, CoordArmPas);
    if fbArmAtiv then
        RotacaoDePontos(falfatemp, CoordArmAtivT, CoordArmAtiv);
    PontosMaximoMinimo(CoordSec, fvmax, fvmin);
    fhalfa:=fvmax-fvmin;
    if fbArmPas then PontosMaximoMinimo(CoordArmPas, fAvmax, fAvmin);
    if fbArmPas then fdalfa:=fAvmax-fAvmin;
    fNdtemp:=0;
    // Arredondar os epsilon, pois eles estão adquirindo erros
    (nas casas decimais avançadas
        fStrtemp:=FloattostrF(fEpsilon1, ffGeneral, 8, 6);
        fEpsilon1:=StrToFloat(fStrtemp);
        fStrtemp:=FloattostrF(fEpsilon2, ffGeneral, 8, 6);
        fEpsilon2:=StrToFloat(fStrtemp);
        if abs(fEpsilon1)<0.000000001 then fEpsilon1:=0;
        if abs(fEpsilon2)<0.000000001 then fEpsilon2:=0;

        if fEpsilon1=fEpsilon2 then begin
            fxEpsilonC2:=10e6;
            fxalfa:=fxEpsilonC2*fEpsilon1/fEpsilonC2;
        end;
        if fEpsilon1<>fEpsilon2 then begin
            fxalfa:=fhalfa*fEpsilon1/(fEpsilon1-fEpsilon2);
            fxEpsilonC2:=fhalfa*fEpsilonC2/(fEpsilon1-fEpsilon2);
//Explicacoes na secao 4.1.2
        end;

fNdtemp:=ForcaConcreto(CoordSec, ffcd, fxalfa, fxEpsilonC2, fNTensaoConcreto);

    if fbArmPas then
        begin
            for fTemp:=0 to Length(CoordArmPasT)-1 do // Inicio
força do aço

```

```

begin
    ahsi[fTemp]:=fvmax-CoordArmPas[fTemp].yi;
    aEpsilonpsi[fTemp]:=fEpsilon1-
ahsi[fTemp]*(fEpsilon1-fEpsilon2)/fhalfa;

fNdtemp:=fNdtemp+aAs[fTemp]*TensaoArmPas(aEpsilonpsi[fTemp],fEs,ffyd);
    end; //fim força do aço
end;
if fbArmAtiv then
begin
for fTemp:=0 to Length(CoordArmAtivT)-1 do // Inicio
força do aço
begin
ahpi[fTemp]:=fvmax-CoordArmAtiv[fTemp].yi;
aEpsilonpsi[fTemp]:=fEpsilon1-
ahpi[fTemp]*(fEpsilon1-fEpsilon2)/fhalfa;

varTemp:=aEpsilonpsi[fTemp]+aEpsilonprealong[fTemp];

fNdtemp:=fNdtemp+aAp[fTemp]*TensaoArmAtiv(fEpsilonpyd,fEpsilonpu,ffpyd,ffp
d,varTemp);

    end; //fim força do aço
end;

fMtempx:=0;
fMtempy:=0;

fMtempx:=MomentoXXConcreto(CoordSec,ffcd,fxalfa,fxEpsilonC2,fNTensaoConcret
o);
    //Primeiro, calculando Mxx

fMtempy:=MomentoYYConcreto(CoordSec,ffcd,fxalfa,fxEpsilonC2,fNTensaoConcret
o);
    //Entao, calculando Myy
if fbArmPas then
begin
for fTemp:=0 to Length(CoordArmPas)-1 do // Inicio
momento do aço
begin
ahsi[fTemp]:=fvmax-CoordArmPas[fTemp].yi;
aEpsilonpsi[fTemp]:=fEpsilon1-
ahsi[fTemp]*(fEpsilon1-fEpsilon2)/fhalfa;

fvarMtemp:=aAs[fTemp]*TensaoArmPas(aEpsilonpsi[fTemp],fEs,ffyd); // Calcula-
se a força na barra e ela é salva numa variavel temporaria

fMtempx:=fMtempx+fvarMtemp*CoordArmPas[fTemp].yi;

fMtempy:=fMtempy+fvarMtemp*CoordArmPas[fTemp].xi;
    end; //fim do momento do aço
end;

if fbArmAtiv then
begin
for fTemp:=0 to Length(CoordArmAtivT)-1 do // Inicio
força da armadura ativa
begin
ahpi[fTemp]:=fvmax-CoordArmAtiv[fTemp].yi;
aEpsilonpsi[fTemp]:=fEpsilon1-
ahpi[fTemp]*(fEpsilon1-fEpsilon2)/fhalfa;

```

```

varTemp:=aEpsilonpsi[fTemp]+aEpsilonprealong[fTemp];

fvarMtemp:=aAp[fTemp]*TensaoArmAtiv(fEpsilonpyd,fEpsilonpu,ffpyd,ffptd,varT
emp); // Calcula-se a força na barra e ela é salva numa variavel temporaria

fMtempx:=fMtemp+fvarMtemp*CoordArmAtiv[fTemp].yi;

fMtempy:=fMtemp+fvarMtemp*CoordArmAtiv[fTemp].xi;
      end; //fim força da armadura ativa
      end;

      if abs(fMtempx)<0.000001 then fMtempx:=0;
      if abs(fMtempy)<0.000001 then fMtempy:=0;

      fMx:=fMtempx*cos(-
falfatemp*3.14159265359/180)+fMtempy*sin(-falfatemp*3.14159265359/180);
//Girando de U para X
      fMy:=-fMtempx*sin(-
falfatemp*3.14159265359/180)+fMtempy*cos(-falfatemp*3.14159265359/180);
//Girando de V para Y

      if abs(fMx)<0.000001 then fMx:=0;
      if abs(fMy)<0.000001 then fMy:=0;

      //Procurando angulo de Mxx e Myy
      if fMx<>0 then
      begin

falfatemp:=arctan(fMy/fMx)*180/3.14159265359;
      end;
      if fMx<0 then falfatemp:=falfatemp+180;
      if fMx>0 then
      begin
      if fMy<0 then
      begin
      falfatemp:=falfatemp+360;
      end;
      end;
      end;

EdtProcResult.Text:='Força='+FloattostrF(fNdtemp,ffGeneral,6,5)+'; Mxx =
'+FloattostrF(fMx,ffGeneral,6,5)+'; Myy =
'+FloattostrF(fMy,ffGeneral,6,5)+'; Angulo do Vetor Momento
='+FloatToStrF(falfatemp,ffGeneral,6,5);
      end;
      end;
end;

```

## B.5 ROTINAS

Nesta seção se apresenta as linhas de programação referentes ao arquivo citado na seção 0, o arquivo *leoufscarfoc*. As rotinas são descritas em seções específicas, pelas sub-seções dentro da seção 5.1.

```
unit leoufscarfoc;
```

```

{$mode objfpc}{$H+}

interface

type
  Tcoord = Record
    npol:integer;
    xi:real;
    yi:real;
  end;

function Ncd1(x1,y1,x2,y2,ymax,x,y,fcd,xEpsilonC2,ffcnNconcreto:Real):Real;
function
Mcd1xx(x1,y1,x2,y2,ymax,x,y,fcd,xEpsilonC2,ffcnNconcreto:Real):Real;
function
Mcd1yy(x1,y1,x2,y2,ymax,x,y,fcd,xEpsilonC2,ffcnNconcreto:Real):Real;
function Ncd2(x1,y1,x2,y2,y,fcd:Real):Real;
function Mcd2xx(x1,y1,x2,y2,y,fcd:Real):Real;
function Mcd2yy(x1,y1,x2,y2,y,fcd:Real):Real;
procedure ObterCaractGeomet(const Coord:array of Tcoord; var fpAc,fpxcg,
fpycg:real);
procedure TranslacaoDePontos(const Coord:array of Tcoord; var
fpxcg,fpycg:real;var CoordT:array of Tcoord);
procedure RotacaoDePontos(const Angulo:real;Coord:array of Tcoord; var
CoordR:array of TCoord);
procedure PontosMaximoMinimo(const Coord:array of Tcoord;var
fpvmax,fpvmin:real);
function TensaoArmPas(var varEpsilonS,varEs,varfyd:Real):real;
function TensaoArmAtiv(var
varEpsilonPu,varEpsilonPu,varfpyd,varfptd,varEpsilonT:Real):real;
function ForcaConcreto(Poligono:array of
Tcoord;ffcnfcd,ffcnxalfa,ffcnxEpsilonC2,ffcnNconcreto:real):real;
function MomentoXXConcreto(Poligono:array of
Tcoord;ffcnfcd,ffcnxalfa,ffcnxEpsilonC2,ffcnNconcreto:real):real;
function MomentoYYConcreto(Poligono:array of
Tcoord;ffcnfcd,ffcnxalfa,ffcnxEpsilonC2,ffcnNconcreto:real):real;

implementation

function Ncd1(x1,y1,x2,y2,ymax,x,y,fcd,xEpsilonC2,ffcnNconcreto:Real):Real;
var
ffuncc,ffuncd,ffunce:real;

begin
  if y1<>y2 then // Todas as condições já excluem a chance de y1=y2, aqui é
possível criar uma faixa de segurança, y1/y2>0,0001, por ex
  begin
    ffuncc:=((y1*x2-y2*x1)/(y1-y2));
    ffuncd:=((x1-x2)/(y1-y2));
  end;
  if y1=y2 then // Todas as condições já excluem a chance de y1=y2, mesmo
assim, é melhor prevenir, criando uma certa segurança y1/y2>0,0001, por ex
  begin
    ffuncc:=0;
    ffuncd:=0;
  end;
  if (((ymax-x)+xEpsilonC2-y)/xEpsilonC2)<=0 then ffunce:=0; // Isto é,
logaritmo neperiano de um numero extremamente pequeno - mas maior e
diferente de zero!

```

```

    if (((ymax-x)+xEpsilonC2-y)/xEpsilonC2)>0 then ffunce:=exp(ln(((ymax-x)+xEpsilonC2-y)/xEpsilonC2)*(ffcnNconcreto+1)); // Rodar a função!

```

```

    Result:=-0.85*fcd*(-
(xEpsilonC2*ffuncc*(ffuncc*(ffcnNconcreto+2)+ffuncc*((ymax-x)+xEpsilonC2+ffcnNconcreto*y+y))
/((ffcnNconcreto+1)*(ffcnNconcreto+2))
-ffuncc*y-(ffuncc*y*y)/2);
end;

```

```

function
Mcd1xx(x1,y1,x2,y2,ymax,x,y,fcd,xEpsilonC2,ffcnNconcreto:Real):Real;

```

```

var
ffuncc,ffuncc,ffuncc:real;

```

```

begin

```

```

    if y1<>y2 then // Todas as condições já excluem a chance de y1=y2, aqui é possível criar uma faixa de segurança, y1/y2>0,0001, por ex

```

```

        begin
            ffuncc:=((y1*x2-y2*x1)/(y1-y2));
            ffuncc:=((x1-x2)/(y1-y2));

```

```

        end;

```

```

    if y1=y2 then // Todas as condições já excluem a chance de y1=y2, aqui é possível criar uma faixa de segurança, y1/y2>0,0001, por ex

```

```

        begin
            ffuncc:=0;
            ffuncc:=0;

```

```

        end;

```

```

    if (((ymax-x)+xEpsilonC2-y)/xEpsilonC2)<=0 then ffuncc:=0; // Isto é, logaritmo neperiano de um numero extremamente pequeno - mas maior e diferente de zero!

```

```

    if (((ymax-x)+xEpsilonC2-y)/xEpsilonC2)>0 then ffuncc:=exp(ln(((ymax-x)+xEpsilonC2-y)/xEpsilonC2)*(ffcnNconcreto+1)); // Rodar a função!

```

```

    Result:=(0.85*fcd*(3*ffuncc*((1+ffcnNconcreto)*(2+ffcnNconcreto))*(3+ffcnNconcreto)*y*y+2*xEpsilonC2*ffuncc*((ymax-x)+xEpsilonC2)*

```

```

(3+ffcnNconcreto)+(3+4*ffcnNconcreto+ffcnNconcreto*ffcnNconcreto)*y))+

```

```

2*ffuncc*((1+ffcnNconcreto)*(2+ffcnNconcreto)*(3+ffcnNconcreto)*y*y*y+
3*xEpsilonC2*ffuncc*(2*(ymax-x)*(ymax-x)+2*xEpsilonC2*xEpsilonC2+
2*xEpsilonC2*(1+ffcnNconcreto)*y+
(2+3*ffcnNconcreto+ffcnNconcreto*ffcnNconcreto)*y*y
+2*(ymax-x)*(2*xEpsilonC2+(1+ffcnNconcreto)*y))))
/(6*(1+ffcnNconcreto)*(2+ffcnNconcreto)*(3+ffcnNconcreto))
;

```

```

end;

```

```

function
Mcd1yy(x1,y1,x2,y2,ymax,x,y,fcd,xEpsilonC2,ffcnNconcreto:Real):Real;

```

```

var
ffuncc,ffuncc,ffuncc:real;

```

```

begin

```

```

    if y1<>y2 then // Todas as condições já excluem a chance de y1=y2, aqui
    é possível criar uma faixa de segurança, y1/y2>0,0001, por ex
    begin
        ffuncc:=((y1*x2-y2*x1)/(y1-y2));
        ffuncd:=((x1-x2)/(y1-y2));
    end;
    if y1=y2 then // Todas as condições já excluem a chance de y1=y2, aqui é
    possível criar uma faixa de segurança, y1/y2>0,0001, por ex
    begin
        ffuncc:=0;
        ffuncd:=0;
    end;
    if ((ymax-x)+xEpsilonC2-y)/xEpsilonC2<=0 then ffuncc:=0; // Isto é,
    logaritmo neperiano de um numero extremamente pequeno - mas maior e
    diferente de zero!
    if (((ymax-x)+xEpsilonC2-y)/xEpsilonC2)>0 then ffuncc:=exp(ln(((ymax-
    x)+xEpsilonC2-y)/xEpsilonC2)*(ffcnNconcreto+1)); // Rodar a função!

    Result:=(0.85*fcd*(3*ffuncc*ffuncc*(xEpsilonC2*
    (6+5*ffcnNconcreto+ffcnNconcreto*ffcnNconcreto)*ffuncc+
    (1+ffcnNconcreto)*(2+ffcnNconcreto)*(3+ffcnNconcreto)*y)+
    3*ffuncc*ffuncd*((1+ffcnNconcreto)*(2+ffcnNconcreto)*(3+ffcnNconcreto)*y*y+
    2*xEpsilonC2*ffuncc*((ymax-x)+xEpsilonC2)*(3+ffcnNconcreto)+
    (3+4*ffcnNconcreto+ffcnNconcreto*ffcnNconcreto)*y))+
    ffuncd*ffuncc*((1+ffcnNconcreto)*(2+ffcnNconcreto)*(3+ffcnNconcreto)*y*y*y+
    3*xEpsilonC2*ffuncc*(2*(ymax-x)*(ymax-x)+
    2*xEpsilonC2*xEpsilonC2+2*xEpsilonC2*(1+ffcnNconcreto)*y+
    (2+3*ffcnNconcreto+ffcnNconcreto*ffcnNconcreto)*y*y+
    2*(ymax-x)*(2*xEpsilonC2+(1+ffcnNconcreto)*y))))/
    (6*(1+ffcnNconcreto)*(2+ffcnNconcreto)*(3+ffcnNconcreto))

    ;

end;

function Ncd2(x1,y1,x2,y2,y,fcd:Real):Real;
var
ffuncc,ffuncd:real;

begin

    if y1<>y2 then // Todas as condições já excluem a chance de y1=y2, aqui
    é possível criar uma faixa de segurança, y1/y2>0,0001, por ex
    begin
        ffuncc:=((y1*x2-y2*x1)/(y1-y2));
        ffuncd:=((x1-x2)/(y1-y2));
    end;
    if y1=y2 then // Todas as condições já excluem a chance de y1=y2, aqui é
    possível criar uma faixa de segurança, y1/y2>0,0001, por ex
    begin
        ffuncc:=0;
        ffuncd:=0;
    end;
    Result:=0.85*fcd*y*(ffuncc+ffuncd*y/2);

end;

```

```

function Mcd2xx(x1,y1,x2,y2,y,fcd:Real):Real;
var
ffuncc,ffuncd:real;

begin

    if y1<>y2 then // Todas as condições já excluem a chance de y1=y2, aqui
é possível criar uma faixa de segurança, y1/y2>0,0001, por ex
        begin
            ffuncc:=((y1*x2-y2*x1)/(y1-y2));
            ffuncd:=((x1-x2)/(y1-y2));
        end;
    if y1=y2 then // Todas as condições já excluem a chance de y1=y2, aqui é
possível criar uma faixa de segurança, y1/y2>0,0001, por ex
        begin
            ffuncc:=0;
            ffuncd:=0;
        end;
    Result:=0.85*fcd*(ffuncc*y*y/2+ffuncd*y*y*y/3)
;

```

```
end;
```

```

function Mcd2yy(x1,y1,x2,y2,y,fcd:Real):Real;
var
ffuncc,ffuncd:real;

```

```
begin
```

```

    if y1<>y2 then // Todas as condições já excluem a chance de y1=y2, aqui
é possível criar uma faixa de segurança, y1/y2>0,0001, por ex
        begin
            ffuncc:=((y1*x2-y2*x1)/(y1-y2));
            ffuncd:=((x1-x2)/(y1-y2));
        end;
    if y1=y2 then // Todas as condições já excluem a chance de y1=y2, aqui é
possível criar uma faixa de segurança, y1/y2>0,0001, por ex
        begin
            ffuncc:=0;
            ffuncd:=0;
        end;

```

```

Result:=0.85*fcd*(ffuncc*ffuncc*y+ffuncc*ffuncd*y*y+ffuncd*ffuncd*y*y*y/3)/
2
;

```

```
end;
```

```

procedure ObterCaractGeomet(const Coord:array of Tcoord; var fpAc,fpxcg,
fpycg:real);
var
fSxx,fSyy:real;
fContador:integer;

```

```
begin
```

```

    fpAc:=0;
    fSxx:=0;
    fSyy:=0;

```

```

    for fContador:=0 to Length(Coord)-2 do
    begin
        fpAc:=fpAc+Coord[fContador].xi*Coord[fContador+1].yi-
Coord[fContador+1].xi*Coord[fContador].yi;
        fSxx:=fSxx+(Coord[fContador].xi*Coord[fContador+1].yi-
Coord[fContador+1].xi*Coord[fContador].yi)*(Coord[fContador].yi+Coord[fContador+1].yi);
        fSyy:=fSyy+(Coord[fContador].xi*Coord[fContador+1].yi-
Coord[fContador+1].xi*Coord[fContador].yi)*(Coord[fContador].xi+Coord[fContador+1].xi);
        end;

        fpAc:=fpAc/2;
        fSxx:=fSxx/6;
        fSyy:=fSyy/6;
        fpxcg:=fSyy/fpAc;
        fpycg:=fSxx/fpAc;
    end;

procedure TranslacaoDePontos(const Coord:array of TCoord; var
fpxcg,fpycg:real;var CoordT:array of TCoord);
var
    fContador:integer;

begin
    for fContador:=0 to Length(Coord)-1 do
    begin
        CoordT[fContador].xi:=Coord[fContador].xi-fpxcg;
        CoordT[fContador].yi:=Coord[fContador].yi-fpycg;
        CoordT[fContador].npol:=Coord[fContador].npol;

        end;
    end;

procedure RotacaoDePontos(const Angulo:real;Coord:array of TCoord; var
CoordR:array of TCoord);
var
    iContador:integer;

    begin
        for iContador:=0 to Length(Coord)-1 do
        begin
            CoordR[iContador].xi:=Coord[iContador].xi*cos(Angulo*pi/180)-
Coord[iContador].yi*sin(Angulo*pi/180);

CoordR[iContador].yi:=Coord[iContador].xi*sin(Angulo*pi/180)+Coord[iContador].yi*cos(Angulo*pi/180);
            CoordR[iContador].npol:=Coord[iContador].npol;
        end;

        end;

procedure PontosMaximoMinimo(const Coord:array of TCoord;var
fpvmax,fpvmin:real);
var
    iContador:integer;

    begin
        fpvmax:=Coord[0].yi;
        fpvmin:=Coord[0].yi;
        for iContador:=1 to Length(Coord)-1 do

```

```

begin
  if Coord[iContador].yi>fpvmax then
    fpvmax:=Coord[iContador].yi;
  if Coord[iContador].yi<fpvmin then
    fpvmin:=Coord[iContador].yi;
end;
end;

function TensaoArmPas(var varEpsilonS,varEs,varfyd:Real):real;
begin
  if varEpsilonS<=varfyd/varEs*1000 then //Trabalhando apenas antes da
ruptura por compressão
  begin
    TensaoArmPas:=-varfyd;
    if varEpsilonS>=-varfyd/varEs*1000 then
      begin
        TensaoArmPas:=varEs*varEpsilonS/1000;
      end;
    end;

    if varEpsilonS>=varfyd/varEs*1000 then
      begin
        TensaoArmPas:=varfyd;
      end;
end;

function TensaoArmAtiv(var
varEpsilononpyd,varEpsilononpu,varfpyd,varfptd,varEpsilont:Real):real;

begin
  Result:=0;
  if varEpsilont<0 then //Lembrar: negativo e tração; e positivo compressao
  begin
    if varEpsilont>=varEpsilononpyd then
      begin
        Result:=varEpsilont*varfpyd/varEpsilononpyd;
      end;
    if varEpsilont<varEpsilononpyd then
      begin
        Result:=varfpyd+(varfptd-varfpyd)/(varEpsilononpu-
varEpsilononpyd)*(varEpsilont-varEpsilononpyd);
      end;
    end;
end;

function ForcaConcreto(Poligono:array of
Tcoord;ffcnfcd,ffcnxalfa,ffcnxEpsilonC2,ffcnNconcreto:real):real;
var
fncnConcContador:integer;
ffcnvmax,ffcnvmin:real;

begin
  Result:=0;
  ffcnvmax:=0;
  ffcnvmin:=0;
  PontosMaximoMinimo(Poligono,ffcnvmax,ffcnvmin);
  for fcnConcContador:=0 to Length(Poligono)-2 do // Inicio do calculo das
forças no concreto
  begin

```

```

        if abs (Poligono [fcnConcContador].yi-
Poligono [fcnConcContador+1].yi)>0.000001 then // Medida para evitar travar
o programa com (Y1 = Y2) que acarreta ( Denominador = Zero )
            begin
                if
Poligono [fcnConcContador].yi<Poligono [fcnConcContador+1].yi then //
Primeiro ponto MENOR que segundo

                    begin
                        if
Poligono [fcnConcContador].yi<ffcnvmax-ffcnxalfa then //Primeiro Ponto Em
tração

                            begin
                                if
Poligono [fcnConcContador+1].yi<=ffcnvmax-ffcnxalfa then
                                    begin
                                        // Se segundo ponto em
tração, fazer nada

                                            end;

                                                if
Poligono [fcnConcContador+1].yi>ffcnvmax-ffcnxalfa then // Segundo ponto
numa área comprimida

                                                    begin
                                                        if
Poligono [fcnConcContador+1].yi<=ffcnvmax-ffcnxalfa+ffcnxEpsilonC2 then //
Segundo ponto na denominada area 1

                                                            begin
                                                                Result:=Result

+Ncd1 (Poligono [fcnConcContador].xi, Poligono [fcnConcContador].yi, Poligono [fc
nConcContador+1].xi, Poligono [fcnConcContador+1].yi, ffcnvmax, ffcnxalfa, Polig
ono [fcnConcContador+1].yi, ffcnfcd, ffcnxEpsilonC2, ffcnNconcreto)
-
Ncd1 (Poligono [fcnConcContador].xi, Poligono [fcnConcContador].yi, Poligono [fc
nConcContador+1].xi, Poligono [fcnConcContador+1].yi, ffcnvmax, ffcnxalfa, ffcnm
ax-ffcnxalfa, ffcnfcd, ffcnxEpsilonC2, ffcnNconcreto);
                                end;
                            if
Poligono [fcnConcContador+1].yi>ffcnvmax-ffcnxalfa+ffcnxEpsilonC2 then //
Segundo ponto na denominada area de compressão 2

                                    begin
                                        Result:=Result

+Ncd1 (Poligono [fcnConcContador].xi, Poligono [fcnConcContador].yi, Poligono [fc
nConcContador+1].xi, Poligono [fcnConcContador+1].yi, ffcnvmax, ffcnxalfa, ffcnv
max-ffcnxalfa+ffcnxEpsilonC2, ffcnfcd, ffcnxEpsilonC2, ffcnNconcreto)
-
Ncd1 (Poligono [fcnConcContador].xi, Poligono [fcnConcContador].yi, Poligono [fc
nConcContador+1].xi, Poligono [fcnConcContador+1].yi, ffcnvmax, ffcnxalfa, ffcnm
ax-ffcnxalfa, ffcnfcd, ffcnxEpsilonC2, ffcnNconcreto)

+Ncd2 (Poligono [fcnConcContador].xi, Poligono [fcnConcContador].yi, Poligono [fc
nConcContador+1].xi, Poligono [fcnConcContador+1].yi, Poligono [fcnConcContador
+1].yi, ffcnfcd)
-
Ncd2 (Poligono [fcnConcContador].xi, Poligono [fcnConcContador].yi, Poligono [fc
nConcContador+1].xi, Poligono [fcnConcContador+1].yi, ffcnvmax-
ffcnxalfa+ffcnxEpsilonC2, ffcnfcd);

```

```

end;
end;

end;
//Primeiro Ponto na primeira área de
compressão
if
Poligono[fcnConcContador].yi<ffcnvmax-ffcnxalfa+ffcnxEpsilonC2 then
begin
if
Poligono[fcnConcContador].yi>=ffcnvmax-ffcnxalfa then
begin
if
Poligono[fcnConcContador+1].yi<=ffcnvmax-ffcnxalfa+ffcnxEpsilonC2 then //
Segundo ponto na denominada area 1
begin
Result:=Result

+Ncd1 (Poligono[fcnConcContador].xi, Poligono[fcnConcContador].yi, Poligono[fc
nConcContador+1].xi, Poligono[fcnConcContador+1].yi, ffcnvmax, ffcnxalfa, Poligo
no[fcnConcContador+1].yi, ffcnfcd, ffcnxEpsilonC2, ffcnNconcreto)
-
Ncd1 (Poligono[fcnConcContador].xi, Poligono[fcnConcContador].yi, Poligono[fcn
ConcContador+1].xi, Poligono[fcnConcContador+1].yi, ffcnvmax, ffcnxalfa, Poligo
no[fcnConcContador].yi, ffcnfcd, ffcnxEpsilonC2, ffcnNconcreto);
end;

if
Poligono[fcnConcContador+1].yi>ffcnvmax-ffcnxalfa+ffcnxEpsilonC2 then //
Segundo ponto na denominada area 2
begin
Result:=Result

+Ncd1 (Poligono[fcnConcContador].xi, Poligono[fcnConcContador].yi, Poligono[fc
nConcContador+1].xi, Poligono[fcnConcContador+1].yi, ffcnvmax, ffcnxalfa, ffcnv
max-ffcnxalfa+ffcnxEpsilonC2, ffcnfcd, ffcnxEpsilonC2, ffcnNconcreto)
-
Ncd1 (Poligono[fcnConcContador].xi, Poligono[fcnConcContador].yi, Poligono[fcn
ConcContador+1].xi, Poligono[fcnConcContador+1].yi, ffcnvmax, ffcnxalfa, Poligo
no[fcnConcContador].yi, ffcnfcd, ffcnxEpsilonC2, ffcnNconcreto)

+Ncd2 (Poligono[fcnConcContador].xi, Poligono[fcnConcContador].yi, Poligono[fc
nConcContador+1].xi, Poligono[fcnConcContador+1].yi, Poligono[fcnConcContador
+1].yi, ffcnfcd)
-
Ncd2 (Poligono[fcnConcContador].xi, Poligono[fcnConcContador].yi, Poligono[fcn
ConcContador+1].xi, Poligono[fcnConcContador+1].yi, ffcnvmax-
ffcnxalfa+ffcnxEpsilonC2, ffcnfcd);
end;

end;

end;
end;

if Poligono[fcnConcContador].yi>=ffcnvmax-
ffcnxalfa+ffcnxEpsilonC2 then//Primeiro Ponto na segunda área de compressão
begin

```

```

Result:=Result

+Ncd2 (Poligono[fcnConcContador].xi, Poligono[fcnConcContador].yi, Poligono[fc
nConcContador+1].xi, Poligono[fcnConcContador+1].yi, Poligono[fcnConcContador
+1].yi, ffcnfcfcd)
-
Ncd2 (Poligono[fcnConcContador].xi, Poligono[fcnConcContador].yi, Poligono[fcn
ConcContador+1].xi, Poligono[fcnConcContador+1].yi, Poligono[fcnConcContador]
.yi, ffcnfcfcd);

end;

end; // Fim de (Primeiro ponto Menor que
segundo ponto)

if
Poligono[fcnConcContador+1].yi<Poligono[fcnConcContador].yi then // Segundo
ponto MENOR que primeiro ponto
begin
if Poligono[fcnConcContador].yi<=ffcnvmax-
ffcnxalfa then //Primeiro Ponto Em tração
begin
// Nada a fazer
end;

if
Poligono[fcnConcContador].yi<=ffcnvmax-ffcnxalfa+ffcnxEpsilonC2 then
//Primeiro Ponto na primeira área de compressão
begin
if
Poligono[fcnConcContador].yi>ffcnvmax-ffcnxalfa then //Primeiro ponto na
primeira area de compressão - condição 2
begin
if
Poligono[fcnConcContador+1].yi>ffcnvmax-ffcnxalfa then // Segundo ponto na
denominada area 1
begin
Result:=Result

+Ncd1 (Poligono[fcnConcContador].xi, Poligono[fcnConcContador].yi, Poligono[fc
nConcContador+1].xi, Poligono[fcnConcContador+1].yi, ffcnvmax, ffcnxalfa, Polig
ono[fcnConcContador+1].yi, ffcnfcfcd, ffcnxEpsilonC2, ffcnNconcreto)
-
Ncd1 (Poligono[fcnConcContador].xi, Poligono[fcnConcContador].yi, Poligono[fcn
ConcContador+1].xi, Poligono[fcnConcContador+1].yi, ffcnvmax, ffcnxalfa, Poligo
no[fcnConcContador].yi, ffcnfcfcd, ffcnxEpsilonC2, ffcnNconcreto);
end;

if
Poligono[fcnConcContador+1].yi<=ffcnvmax-ffcnxalfa then // Segundo ponto na
area de tração
begin
Result:=Result

+Ncd1 (Poligono[fcnConcContador].xi, Poligono[fcnConcContador].yi, Poligono[fc
nConcContador+1].xi, Poligono[fcnConcContador+1].yi, ffcnvmax, ffcnxalfa, ffcnv
max-ffcnxalfa, ffcnfcfcd, ffcnxEpsilonC2, ffcnNconcreto)

```

```

-
Ncd1 (Poligono[fcnConcContador].xi, Poligono[fcnConcContador].yi, Poligono[fcnConcContador+1].xi, Poligono[fcnConcContador+1].yi, ffcnvmax, ffcnxalfa, Poligono[fcnConcContador].yi, ffcnfcd, ffcnxEpsilonC2, ffcnNconcreto);
end;

end;
end;

if Poligono[fcnConcContador].yi > ffcnvmax - ffcnxalfa + ffcnxEpsilonC2 then //Primeiro Ponto na segunda área de compressão
begin
if
Poligono[fcnConcContador+1].yi >= ffcnvmax - ffcnxalfa + ffcnxEpsilonC2 then
//Segundo ponto na segunda área de compressão
begin
Result := Result

+Ncd2 (Poligono[fcnConcContador].xi, Poligono[fcnConcContador].yi, Poligono[fcnConcContador+1].xi, Poligono[fcnConcContador+1].yi, Poligono[fcnConcContador+1].yi, ffcnfcd)
-
Ncd2 (Poligono[fcnConcContador].xi, Poligono[fcnConcContador].yi, Poligono[fcnConcContador+1].xi, Poligono[fcnConcContador+1].yi, Poligono[fcnConcContador].yi, ffcnfcd);

end;

if
Poligono[fcnConcContador+1].yi < ffcnvmax - ffcnxalfa + ffcnxEpsilonC2 then
//Segundo ponto na primeira área de compressão
begin
if
Poligono[fcnConcContador+1].yi >= ffcnvmax - ffcnxalfa then //Segundo ponto na primeira área de compressão - condição 2
begin
Result := Result

+Ncd1 (Poligono[fcnConcContador].xi, Poligono[fcnConcContador].yi, Poligono[fcnConcContador+1].xi, Poligono[fcnConcContador+1].yi, Poligono[fcnConcContador+1].yi, ffcnvmax, ffcnxalfa, Poligono[fcnConcContador+1].yi, ffcnfcd, ffcnxEpsilonC2, ffcnNconcreto)
-
Ncd1 (Poligono[fcnConcContador].xi, Poligono[fcnConcContador].yi, Poligono[fcnConcContador+1].xi, Poligono[fcnConcContador+1].yi, Poligono[fcnConcContador+1].yi, ffcnvmax, ffcnxalfa, ffcnvmax - ffcnxalfa + ffcnxEpsilonC2, ffcnfcd, ffcnxEpsilonC2, ffcnNconcreto)

+Ncd2 (Poligono[fcnConcContador].xi, Poligono[fcnConcContador].yi, Poligono[fcnConcContador+1].xi, Poligono[fcnConcContador+1].yi, Poligono[fcnConcContador+1].yi, ffcnvmax - ffcnxalfa + ffcnxEpsilonC2, ffcnfcd)
-
Ncd2 (Poligono[fcnConcContador].xi, Poligono[fcnConcContador].yi, Poligono[fcnConcContador+1].xi, Poligono[fcnConcContador+1].yi, Poligono[fcnConcContador].yi, ffcnfcd);

end;
end;

```

```

                                if
Poligono[fcnConcContador+1].yi<ffcnvmax-ffcnxalfa then // Segundo ponto na
area tracionada
                                begin
                                    Result:=Result

+Ncd1 (Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[fc
nConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax,ffcnxalfa,ffcnv
max-ffcnxalfa,ffcnfcd,ffcnxEpsilonC2,ffcnNconcreto)
-
Ncd1 (Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[fc
nConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax,ffcnxalfa,ffcnv
max-ffcnxalfa+ffcnxEpsilonC2,ffcnfcd,ffcnxEpsilonC2,ffcnNconcreto)

+Ncd2 (Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[fc
nConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax-
ffcnxalfa+ffcnxEpsilonC2,ffcnfcd)
-
Ncd2 (Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[fc
nConcContador+1].xi,Poligono[fcnConcContador+1].yi,Poligono[fcnConcContador]
.yi,ffcnfcd);
                                end;

                                end;

                                end; // Final dos calculos com segundo
ponto menor que o primeiro para compressão do concreto
                                end;

                                end; // Fim do calculo das forças no concreto
end;
// Força concreto - Fim

function MomentoXXConcreto(Poligono:array of
Tcoord;ffcnfcd,ffcnxalfa,ffcnxEpsilonC2,ffcnNconcreto:real):real;
var
fcnConcContador:integer;
ffcnvmax,ffcnvmin:real;

begin

    Result:=0;
    ffcnvmax:=0;
    ffcnvmin:=0;

    PontosMaximoMinimo(Poligono,ffcnvmax,ffcnvmin);

    for fcnConcContador:=0 to Length(Poligono)-2 do // Inicio do
calculo do momento xx no concreto
        begin
            if abs(Poligono[fcnConcContador].yi-
Poligono[fcnConcContador+1].yi)>0.000001 then // Medida para evitar travar
o programa com (Y1 = Y2) que acarreta ( Denominador = Zero )
                begin

                    if
Poligono[fcnConcContador].yi<Poligono[fcnConcContador+1].yi then //
Primeiro ponto MENOR que segundo

```

```

begin
  if
    Poligono[fcnConcContador].yi<ffcnvmax-ffcnxalfa then //Primeiro Ponto Em
    tração
      begin
        if
          Poligono[fcnConcContador+1].yi<=ffcnvmax-ffcnxalfa then
            begin
              // Se segundo ponto em
              tração, fazer nada
            end;
          if
            Poligono[fcnConcContador+1].yi>ffcnvmax-ffcnxalfa then // Segundo ponto
            numa área comprimida
              begin
                if
                  Poligono[fcnConcContador+1].yi<=ffcnvmax-ffcnxalfa+ffcnxEpsilonC2 then //
                  Segundo ponto na denominada area 1
                    begin
                      Result:=Result
                    +Mcd1xx(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[
                    fcnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax,ffcnxalfa,Pol
                    igono[fcnConcContador+1].yi,ffcnfcd,ffcnxEpsilonC2,ffcnNconcreto)
                    -
                    Mcd1xx(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[f
                    cnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax,ffcnxalfa,ffcn
                    vmax-ffcnxalfa,ffcnfcd,ffcnxEpsilonC2,ffcnNconcreto);
                    end;
                if
                  Poligono[fcnConcContador+1].yi>ffcnvmax-ffcnxalfa+ffcnxEpsilonC2 then //
                  Segundo ponto na denominada area de compressão 2
                    begin
                      Result:=Result
                    +Mcd1xx(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[
                    fcnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax,ffcnxalfa,ffc
                    nvmax-ffcnxalfa+ffcnxEpsilonC2,ffcnfcd,ffcnxEpsilonC2,ffcnNconcreto)
                    -
                    Mcd1xx(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[f
                    cnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax,ffcnxalfa,ffcn
                    vmax-ffcnxalfa,ffcnfcd,ffcnxEpsilonC2,ffcnNconcreto)
                    +Mcd2xx(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[
                    fcnConcContador+1].xi,Poligono[fcnConcContador+1].yi,Poligono[fcnConcContad
                    or+1].yi,ffcnfcd)
                    -
                    Mcd2xx(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[f
                    cnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax-
                    ffcnXalfa+ffcnxEpsilonC2,ffcnfcd);
                    end;
                end;
              end;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

```

//Primeiro Ponto na primeira área de
compressão
        if
Poligono[fcnConcContador].yi<ffcnvmax-ffcnxalfa+ffcnxEpsilonC2 then
            begin
                if
Poligono[fcnConcContador].yi>=ffcnvmax-ffcnxalfa then
                    begin
                        if
Poligono[fcnConcContador+1].yi<=ffcnvmax-ffcnxalfa+ffcnxEpsilonC2 then //
Segundo ponto na denominada area 1
                            begin
                                Result:=Result

+Mcd1xx (Poligono [fcnConcContador] .xi, Poligono [fcnConcContador] .yi, Poligono [
fcnConcContador+1] .xi, Poligono [fcnConcContador+1] .yi, ffcnvmax, ffcnxalfa, Pol
igono [fcnConcContador+1] .yi, ffcnfcd, ffcnxEpsilonC2, ffcnNconcreto)
-
Mcd1xx (Poligono [fcnConcContador] .xi, Poligono [fcnConcContador] .yi, Poligono [f
cnConcContador+1] .xi, Poligono [fcnConcContador+1] .yi, ffcnvmax, ffcnxalfa, Poli
gono [fcnConcContador] .yi, ffcnfcd, ffcnxEpsilonC2, ffcnNconcreto) ;
                                end;

                            if
Poligono[fcnConcContador+1].yi>ffcnvmax-ffcnxalfa+ffcnxEpsilonC2 then //
Segundo ponto na denominada area 2
                                    begin
                                        Result:=Result

+Mcd1xx (Poligono [fcnConcContador] .xi, Poligono [fcnConcContador] .yi, Poligono [
fcnConcContador+1] .xi, Poligono [fcnConcContador+1] .yi, ffcnvmax, ffcnxalfa, ffc
nvmax-ffcnxalfa+ffcnxEpsilonC2, ffcnfcd, ffcnxEpsilonC2, ffcnNconcreto)
-
Mcd1xx (Poligono [fcnConcContador] .xi, Poligono [fcnConcContador] .yi, Poligono [f
cnConcContador+1] .xi, Poligono [fcnConcContador+1] .yi, ffcnvmax, ffcnxalfa, Poli
gono [fcnConcContador] .yi, ffcnfcd, ffcnxEpsilonC2, ffcnNconcreto)

+Mcd2xx (Poligono [fcnConcContador] .xi, Poligono [fcnConcContador] .yi, Poligono [
fcnConcContador+1] .xi, Poligono [fcnConcContador+1] .yi, Poligono [fcnConcContad
or+1] .yi, ffcnfcd)
-
Mcd2xx (Poligono [fcnConcContador] .xi, Poligono [fcnConcContador] .yi, Poligono [f
cnConcContador+1] .xi, Poligono [fcnConcContador+1] .yi, ffcnvmax-
ffcnxalfa+ffcnxEpsilonC2, ffcnfcd) ;
                                    end;

                                end;
                            end;

                        if Poligono[fcnConcContador].yi>=ffcnvmax-
ffcnxalfa+ffcnxEpsilonC2 then//Primeiro Ponto na segunda área de compressão
                            begin
                                Result:=Result

+Mcd2xx (Poligono [fcnConcContador] .xi, Poligono [fcnConcContador] .yi, Poligono [
fcnConcContador+1] .xi, Poligono [fcnConcContador+1] .yi, Poligono [fcnConcContad
or+1] .yi, ffcnfcd)
-
Mcd2xx (Poligono [fcnConcContador] .xi, Poligono [fcnConcContador] .yi, Poligono [f

```



```

end;
end;

if Poligono[fcnConcContador].yi>ffcnvmax-
ffcnxalfa+ffcnxEpsilonC2 then //Primeiro Ponto na segunda área de
compressão
begin
if
Poligono[fcnConcContador+1].yi>=ffcnvmax-ffcnxalfa+ffcnxEpsilonC2 then
//Segundo ponto na segunda área de compressão
begin
Result:=Result

+Mcd2xx(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[
fcnConcContador+1].xi,Poligono[fcnConcContador+1].yi,Poligono[fcnConcContad
or+1].yi,ffcnfcd)
-
Mcd2xx(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[f
cnConcContador+1].xi,Poligono[fcnConcContador+1].yi,Poligono[fcnConcContado
r].yi,ffcnfcd);

end;

if
Poligono[fcnConcContador+1].yi<ffcnvmax-ffcnxalfa+ffcnxEpsilonC2 then
//Segundo ponto na primeira área de compressão
begin
if
Poligono[fcnConcContador+1].yi>=ffcnvmax-ffcnxalfa then //Segundo ponto na
primeira área de compressão - condição 2
begin
Result:=Result

+Mcd1xx(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[
fcnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax,ffcnxalfa,Pol
igono[fcnConcContador+1].yi,ffcnfcd,ffcnxEpsilonC2,ffcnNconcreto)
-
Mcd1xx(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[f
cnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax,ffcnxalfa,ffcn
vmax-ffcnxalfa+ffcnxEpsilonC2,ffcnfcd,ffcnxEpsilonC2,ffcnNconcreto)

+Mcd2xx(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[
fcnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax-
ffcnxalfa+ffcnxEpsilonC2,ffcnfcd)
-
Mcd2xx(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[f
cnConcContador+1].xi,Poligono[fcnConcContador+1].yi,Poligono[fcnConcContado
r].yi,ffcnfcd);

end;
end;

if
Poligono[fcnConcContador+1].yi<ffcnvmax-ffcnxalfa then // Segundo ponto na
area tracionada
begin
Result:=Result

```

```

+Mcd1xx (Poligono[fcnConcContador].xi, Poligono[fcnConcContador].yi, Poligono[
fcnConcContador+1].xi, Poligono[fcnConcContador+1].yi, ffcnvmax, ffcnxalfa, ffc
nvmax-ffcnxalfa, ffcnfcd, ffcnxEpsilonC2, ffcnNconcreto)
-
Mcd1xx (Poligono[fcnConcContador].xi, Poligono[fcnConcContador].yi, Poligono[f
cnConcContador+1].xi, Poligono[fcnConcContador+1].yi, ffcnvmax, ffcnxalfa, ffc
nvmax-ffcnxalfa+ffcnxEpsilonC2, ffcnfcd, ffcnxEpsilonC2, ffcnNconcreto)

+Mcd2xx (Poligono[fcnConcContador].xi, Poligono[fcnConcContador].yi, Poligono[
fcnConcContador+1].xi, Poligono[fcnConcContador+1].yi, ffcnvmax-
ffcnxalfa+ffcnxEpsilonC2, ffcnfcd)
-
Mcd2xx (Poligono[fcnConcContador].xi, Poligono[fcnConcContador].yi, Poligono[f
cnConcContador+1].xi, Poligono[fcnConcContador+1].yi, Poligono[fcnConcContado
r].yi, ffcnfcd);
end;

end;

end; // Final dos calculos com segundo
ponto menor que o primeiro para compressão do concreto
end;
end; // Fim do calculo do momento Mxx no concreto

end;

function MomentoYYConcreto(Poligono:array of
Tcoord; ffcnfcd, ffcnxalfa, ffcnxEpsilonC2, ffcnNconcreto:real):real;
var
fcnConcContador:integer;
ffcnvmax, ffcnvmin:real;
begin
Result:=0;
ffcnvmax:=0;
ffcnvmin:=0;

PontosMaximoMinimo(Poligono, ffcnvmax, ffcnvmin);
for fcnConcContador:=0 to Length(Poligono)-2 do // Inicio do calculo
do momento yy no concreto
begin
if abs(Poligono[fcnConcContador].yi-
Poligono[fcnConcContador+1].yi)>0.000001 then // Medida para evitar travar
o programa com (Y1 = Y2) que acarreta ( Denominador = Zero )
begin
if
Poligono[fcnConcContador].yi<Poligono[fcnConcContador+1].yi then //
Primeiro ponto MENOR que segundo
begin
if Poligono[fcnConcContador].yi<ffcnvmax-
ffcnxalfa then //Primeiro Ponto Em tração
begin
if
Poligono[fcnConcContador+1].yi<=ffcnvmax-ffcnxalfa then
begin

```

```

// Se segundo ponto em tração,
fazer nada
end;

if
Poligono[fcnConcContador+1].yi>ffcnvmax-ffcnxalfa then // Segundo ponto
numa área comprimida
begin
if
Poligono[fcnConcContador+1].yi<=ffcnvmax-ffcnxalfa+ffcnxEpsilonC2 then //
Segundo ponto na denominada area 1
begin
Result:=Result

+Mcd1yy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[
fcnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax,ffcnxalfa,Pol
igono[fcnConcContador+1].yi,ffcnfcd,ffcnxEpsilonC2,ffcnNconcreto)
-
Mcd1yy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[f
cnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax,ffcnxalfa,ffc
nvmx-ffcnxalfa,ffcnfcd,ffcnxEpsilonC2,ffcnNconcreto);
end;
if
Poligono[fcnConcContador+1].yi>ffcnvmax-ffcnxalfa+ffcnxEpsilonC2 then //
Segundo ponto na denominada area de compressão 2
begin
Result:=Result

+Mcd1yy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[
fcnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax,ffcnxalfa,ffc
nvmax-ffcnxalfa+ffcnxEpsilonC2,ffcnfcd,ffcnxEpsilonC2,ffcnNconcreto)
-
Mcd1yy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[f
cnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax,ffcnxalfa,ffc
nvmax-ffcnxalfa,ffcnfcd,ffcnxEpsilonC2,ffcnNconcreto)

+Mcd2yy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[
fcnConcContador+1].xi,Poligono[fcnConcContador+1].yi,Poligono[fcnConcContad
or+1].yi,ffcnfcd)
-
Mcd2yy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[f
cnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax-
ffcnxalfa+ffcnxEpsilonC2,ffcnfcd);
end;
end;

end;
//Primeiro Ponto na primeira área de
compressão
if
Poligono[fcnConcContador].yi<ffcnvmax-ffcnxalfa+ffcnxEpsilonC2 then
begin
if
Poligono[fcnConcContador].yi>=ffcnvmax-ffcnxalfa then
begin

```

```

                                if
Poligono[fcnConcContador+1].yi<=ffcnvmax-ffcnxalfa+ffcnxEpsilonC2 then //
Segundo ponto na denominada area 1
                                begin
                                    Result:=Result

+Mcd1yy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[
fcnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax,ffcnxalfa,Pol
igono[fcnConcContador+1].yi,ffcnfcd,ffcnxEpsilonC2,ffcnNconcreto)
-
Mcd1yy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[f
cnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax,ffcnxalfa,Pol
igono[fcnConcContador].yi,ffcnfcd,ffcnxEpsilonC2,ffcnNconcreto);
                                end;

                                if
Poligono[fcnConcContador+1].yi>ffcnvmax-ffcnxalfa+ffcnxEpsilonC2 then //
Segundo ponto na denominada area 2
                                begin
                                    Result:=Result

+Mcd1yy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[
fcnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax,ffcnxalfa,ffc
nvmax-ffcnxalfa+ffcnxEpsilonC2,ffcnfcd,ffcnxEpsilonC2,ffcnNconcreto)
-
Mcd1yy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[f
cnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax,ffcnxalfa,Pol
igono[fcnConcContador].yi,ffcnfcd,ffcnxEpsilonC2,ffcnNconcreto)

+Mcd2yy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[
fcnConcContador+1].xi,Poligono[fcnConcContador+1].yi,Poligono[fcnConcContad
or+1].yi,ffcnfcd)
-
Mcd2yy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[f
cnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax-
ffcnxalfa+ffcnxEpsilonC2,ffcnfcd);
                                end;

                                end;
                                end;

                                if Poligono[fcnConcContador].yi>=ffcnvmax-
ffcnxalfa+ffcnxEpsilonC2 then//Primeiro Ponto na segunda área de compressão
                                begin
                                    Result:=Result

+Mcd2yy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[
fcnConcContador+1].xi,Poligono[fcnConcContador+1].yi,Poligono[fcnConcContad
or+1].yi,ffcnfcd)
-
Mcd2yy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[f
cnConcContador+1].xi,Poligono[fcnConcContador+1].yi,Poligono[fcnConcContado
r].yi,ffcnfcd);

                                end;

                                end; // Fim de (Primeiro ponto Menor que segundo
ponto)

```

```

        if
Poligono[fcnConcContador+1].yi<Poligono[fcnConcContador].yi then // Segundo
ponto MENOR que primeiro ponto
        begin

                if Poligono[fcnConcContador].yi<=ffcnvmax-
ffcnxalfa then //Primeiro Ponto Em tração
                begin
                        // Nada a fazer
                end;

                if Poligono[fcnConcContador].yi<=ffcnvmax-
ffcnxalfa+ffcnxEpsilonC2 then //Primeiro Ponto na primeira área de
compressão

                begin

                        if
Poligono[fcnConcContador].yi>ffcnvmax-ffcnxalfa then //Primeiro ponto na
primeira area de compressão - condição 2
                        begin

                                if
Poligono[fcnConcContador+1].yi>ffcnvmax-ffcnxalfa then // Segundo ponto na
denominada area 1

                                begin
                                        Result:=Result

+Mcdlyy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[
fcnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax,ffcnxalfa,Pol
igono[fcnConcContador+1].yi,ffcnfcd,ffcnxEpsilonC2,ffcnNconcreto)
-
Mcdlyy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[f
cnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax,ffcnxalfa,Pol
igono[fcnConcContador].yi,ffcnfcd,ffcnxEpsilonC2,ffcnNconcreto);
                                end;

                                if
Poligono[fcnConcContador+1].yi<=ffcnvmax-ffcnxalfa then // Segundo ponto na
area de tração

                                begin
                                        Result:=Result

+Mcdlyy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[
fcnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax,ffcnxalfa,ffc
nvmax-ffcnxalfa,ffcnfcd,ffcnxEpsilonC2,ffcnNconcreto)
-
Mcdlyy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[f
cnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax,ffcnxalfa,Pol
igono[fcnConcContador].yi,ffcnfcd,ffcnxEpsilonC2,ffcnNconcreto);
                                end;

                                end;
                        end;

                if Poligono[fcnConcContador].yi>ffcnvmax-
ffcnxalfa+ffcnxEpsilonC2 then //Primeiro Ponto na segunda área de
compressão

                begin

```

```

                                if
Poligono[fcnConcContador+1].yi>=ffcnvmax-ffcnxalfa+ffcnxEpsilonC2 then
//Segundo ponto na segunda área de compressão
                                begin
                                        Result:=Result

+Mcd2yy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[
fcnConcContador+1].xi,Poligono[fcnConcContador+1].yi,Poligono[fcnConcContad
or+1].yi,ffcnfcd)
                                -
Mcd2yy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[f
cnConcContador+1].xi,Poligono[fcnConcContador+1].yi,Poligono[fcnConcContado
r].yi,ffcnfcd);

                                end;

                                if
Poligono[fcnConcContador+1].yi<ffcnvmax-ffcnxalfa+ffcnxEpsilonC2 then
//Segundo ponto na primeira área de compressão
                                begin
                                        if
Poligono[fcnConcContador+1].yi>=ffcnvmax-ffcnxalfa then //Segundo ponto na
primeira área de compressão - condição 2
                                                begin
                                                        Result:=Result

+Mcd1yy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[
fcnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax,ffcnxalfa,Pol
igono[fcnConcContador+1].yi,ffcnfcd,ffcnxEpsilonC2,ffcnNconcreto)
                                                -
Mcd1yy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[f
cnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax,ffcnxalfa,ffcn
vmax-ffcnxalfa+ffcnxEpsilonC2,ffcnfcd,ffcnxEpsilonC2,ffcnNconcreto)

+Mcd2yy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[
fcnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax-
ffcnxalfa+ffcnxEpsilonC2,ffcnfcd)
                                                -
Mcd2yy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[f
cnConcContador+1].xi,Poligono[fcnConcContador+1].yi,Poligono[fcnConcContado
r].yi,ffcnfcd);

                                                end;
                                end;

                                if
Poligono[fcnConcContador+1].yi<ffcnvmax-ffcnxalfa then // Segundo ponto na
area tracionada
                                begin
                                        Result:=Result

+Mcd1yy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[
fcnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax,ffcnxalfa,ffc
nvmax-ffcnxalfa,ffcnfcd,ffcnxEpsilonC2,ffcnNconcreto)
                                -
Mcd1yy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[f
cnConcContador+1].xi,Poligono[fcnConcContador+1].yi,ffcnvmax,ffcnxalfa,ffcn
vmax-ffcnxalfa+ffcnxEpsilonC2,ffcnfcd,ffcnxEpsilonC2,ffcnNconcreto)

+Mcd2yy(Poligono[fcnConcContador].xi,Poligono[fcnConcContador].yi,Poligono[

```

```

fcncncContador+1].xi,Poligono[fcncncContador+1].yi,fcncncvmax-
fcncncxalfa+fcncncxEpsilonC2,fcncncfcd)
-
Mcd2yy(Poligono[fcncncContador].xi,Poligono[fcncncContador].yi,Poligono[fcncncContador+1].xi,Poligono[fcncncContador+1].yi,Poligono[fcncncContador].xi,Poligono[fcncncContador+1].yi,fcncncfcd);
end;
end;
end; // Final dos calculos com segundo ponto
menor que o primeiro para compressão do concreto
end;
end; // Fim do calculo do momento yy no concreto
end;
end.

```