

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA CIVIL**

Leonardo Costa de Souza

**PROGRAMA DE COMPUTADOR PARA PROJETO DE VIGAS
PROTENDIDAS**

Porto Alegre
Novembro 2016

LEONARDO COSTA DE SOUZA

**PROGRAMA DE COMPUTADOR PARA PROJETO DE VIGAS
PROTENDIDAS**

Trabalho de Diplomação apresentado ao Departamento de Engenharia Civil da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para obtenção do título de Engenheiro Civil

Orientador/a: Rubem Clecio Schwingel

Porto Alegre
Novembro 2016

LEONARDO COSTA DE SOUZA

**PROGRAMA DE COMPUTADOR PARA PROJETO DE VIGAS
PROTENDIDAS**

Este Trabalho de Diplomação foi julgado adequado como pré-requisito para a obtenção do título de ENGENHEIRO CIVIL e aprovado em sua forma final pelo Professor Orientador e pelo Relator da disciplina Trabalho de Conclusão de Curso II - Civil da Universidade Federal do Rio Grande do Sul.

Porto Alegre, novembro de 2016

Prof. Rubem Clecio Schwingel
Msc. pela Universidade Federal do Rio Grande do Sul
Orientador

Prof. Roberto Domingo Rios
Dr. pela Universidade Federal do Rio Grande do Sul
Relator

BANCA EXAMINADORA

Prof. Rubem Clecio Schwingel (UFRGS)
Msc. pela Universidade Federal do Rio Grande do Sul

Prof. Roberto Domingo Rios (UFRGS)
Dr. pela Universidade Federal do Rio Grande do Sul

Prof. Ronald José Ellwanger (UFRGS)
Dr. pela Universidade Federal do Rio de Janeiro

Dedico este trabalho a meus pais, Alexandre e Valdinéa,
aos meus irmãos, Cristiano, Rafael e Frederico, que
sempre me apoiaram e especialmente durante o período do
meu Curso de Graduação estiveram ao meu lado.

AGRADECIMENTOS

Agradeço ao Prof. Rubem Clecio Schwingel, orientador deste trabalho, pela dedicação, colaboração e ensinamentos que foram fundamentais ao desenvolvimento deste trabalho.

Agradeço ao Prof. Roberto Domingo Rios, por suas contribuições, recomendações e críticas valiosas para o aperfeiçoamento deste trabalho.

Agradeço ao Prof. Ronald José Ellwanger, por seus ensinamentos e contribuições para superar dificuldades no desenvolvimento desse trabalho.

Agradeço o Prof. João Ricardo Masuero, pelos conhecimentos passados e valiosa contribuição em apresentar conteúdos além do essencial ao curso, permitindo a elaboração deste trabalho.

Agradeço aos demais professores do curso que sempre tiveram grande dedicação em passar seus conhecimentos e a ensinar da melhor maneira possível o necessário para me formar engenheiro.

Agradeço aos colegas e amigos que me apoiaram durante todo o curso e que foram de importante contribuição para minha formação.

Agradeço aos meus familiares e amigos que sempre me deram suporte para que eu entrasse novamente na faculdade para me graduar engenheiro civil.

С одной логикой нельзя через натуру перескочить.
Логика предугадает три случая, а их миллион.

É simplesmente impossível saltar com lógica por cima da
natureza. A lógica pressupõe três casos, ao passo que há
milhões deles.

Fiódor Dostoiévski

RESUMO

Este trabalho versa sobre a organização e implementação de um programa de computador para o projeto de vigas protendidas. São abordados aspectos tanto do ponto de vista da entrada dos dados e operações de cálculo necessárias à obtenção de resultados, bem como a teoria necessária ao estabelecimento de um fluxo de projeto considerando os elementos para dimensionar uma armadura de protensão por pós-tração em vigas de concreto. Para implementação do programa foi adotada linguagem C# em ambiente de programação Visual Studio 2015, compilado para sistemas operacionais Windows x64 (64 bits). Foi desenvolvido um código que possibilita a entrada de dados de projeto, como dimensões da viga e seção, tipos de materiais utilizados, cargas atuantes sobre a estrutura em função do tempo (idade do concreto), restrições nodais como apoios, vínculos elásticos e deslocamentos pré-determinados, em ambiente gráfico tridimensional. Também foram implementadas rotinas que solucionam a estrutura pelo método da rigidez, montando uma matriz de rigidez da estrutura, um vetor de cargas nos vínculos, obtendo um vetor de deslocamentos e a partir deste as reações dos apoios. Com os resultados obtidos implementou-se então um visualizador gráfico que calcula e exibe as solicitações de cortante e momento de flexão ao longo da estrutura fornecida. No que tange o fluxo de projeto de vigas protendidas, foi realizado um estudo completo das metodologias necessárias à introdução da armadura de protensão, em especial, para vigas hiperestáticas pelo método das cargas equivalentes. Foi definido o procedimento para estabelecer uma melhor geometria para um cabo de protensão composto por segmentos de parábolas, facilmente transformável em cargas equivalentes que podem reutilizar os algoritmos já implementados para obtenção dos resultados ao longo da viga. Por fim foram também analisadas as perdas de protensão pertinentes e apresentadas suas equações necessárias para futura implementação e inserção no código de programa desenvolvido.

Palavras-chave: NBR 6.118; protendido; pós-tração; programação; vigas; software.

LISTA DE FIGURAS

Figura 1 – Diagrama das etapas de pesquisa.	22
Figura 2 – Segmentação da viga em trechos.	33
Figura 3 – Segmentação da armadura de protensão em trechos de parábolas.	34
Figura 4 – Relação de flechas e ângulo do ponto de inflexão entre parábolas.	37
Figura 5 – Parâmetros geométricos da seção da viga por tipo de seção.	41
Figura 6 – Diagrama tensão-deformação idealizado.	47
Figura 7 – Diagrama tensão deformação para aços de armaduras passivas.	49
Figura 8 – Diagrama tensão deformação para aços de armaduras ativas.	50
Figura 9 – Solicitações internas na viga devido à protensão.	53
Figura 10 – Relações trigonométricas com ângulo inicial e forma da parábola.	53
Figura 11 – Definição de uma parábola equivalente para o trecho para simplificação das cargas.	57
Figura 12 – Força de atrito em cabo curvo.	69
Figura 13: Variação angular do cabo de protensão ao longo de uma viga.	70
Figura 14 – Tensões do cabo de protensão antes e após recuo de ancoragem.	73
Figura 15 – Interface principal do programa.	86
Figura 16 – Exemplo de janela de diálogo com formulário para coleta de dados.	87
Figura 17 – Inserção de novo elemento grid ao longo da viga.	89
Figura 18 – Representação de cargas ao longo da viga.	91
Figura 19 - Representação gráfica das restrições conforme parâmetros selecionados.	92
Figura 20 - Representação gráfica das restrições conforme parâmetros selecionados.	93
Figura 21 - Remoção de um instante de tempo da lista de idades do concreto.	94
Figura 22 - Exibição gráfica das condições inseridas para o cálculo de solicitações.	96
Figura 23 – Exibição gráfica das solicitações calculadas pelo programa.	96
Figura 24 – Representação do problema resolvido no exemplo 1.	118
Figura 25 – Representação do problema resolvido no exemplo 2.	119
Figura 26 – Representação do exemplo 3 no ftool.	120
Figura 27 – Representação do exemplo 3 no PSB.	120

LISTA DE TABELAS

Tabela 1 – Fórmulas para determinação das excentricidades de cada parábola.....	36
Tabela 2 – Exigências de durabilidade relacionadas à fissuração e à proteção da armadura, em função das classes de agressividade ambiental.	43
Tabela 3 – Comparação das projeções de valores da protensão por cálculo exato e por simplificação.	55
Tabela 4 – Valores de ψ_{1000} , em porcentagem.....	77
Tabela 5 – Valores numéricos usuais para determinação da fluência e retração.	80
Tabela 6 – Valores de fluência e de retração em função da velocidade de endurecimento do cimento	84
Tabela 7 - Equivalência das cargas nodais pelo método dos deslocamentos.....	112
Tabela 8 – Comparação de resultados para o exemplo 1.	118
Tabela 9 – Comparação de resultados para o exemplo 1.	119
Tabela 10 – Comparação dos diagramas obtidos para as solicitações da viga.....	121
Tabela 11 – Comparação de resultados para o exemplo 3.	121

LISTA DE SIGLAS

UFRGS – Universidade Federal do Rio Grande do Sul

WPF – Windows Presentation Foundation

LISTA DE SÍMBOLOS

{ } – matrizes bidimensionais (n) x (m) são representadas por chaves;

[] – vetores ou matrizes unidimensionais (n) x 1 são representadas por colchetes;

{F} – matriz (n x m) x 1 de forças resultantes para os n nós da estrutura para cada um das m direções:

- a) forças na direção horizontal (kN);
- b) forças na direção vertical (kN);
- c) forças em torno do eixo z (kN.m).

[K] – matriz de rigidez (n x m) x (n x m) da estrutura para cada um dos n nós nas (m x m) direções:

- a) rigidez na direção x-x (kN/m);
- b) rigidez na direção y-y (kN/m);
- c) rigidez na direção y-z (kN/rad);
- d) rigidez em torno de z-z (kN.m/rad);
- e) demais direções foram desconsideradas, pois possuem rigidez nula para esse trabalho.

{U} – matriz de deslocamentos (n x m) x 1 para cada n nós da estrutura para cada uma das m direções:

- a) deslocamento na direção x (m);
- b) deslocamento na direção y (m);
- c) deslocamento em torno do eixo z (rad);

[K^L] – matriz de rigidez local (2x3) x (2x3) para cada elemento barra de nós i e j para cada direção:

- a) rigidez na direção x-x (kN/m);
- b) rigidez na direção y-y (kN/m);
- c) rigidez na direção y-z (kN/rad);
- d) rigidez em torno de z-z (kN.m/rad);

- e) demais direções foram desconsideradas, pois possuem rigidez nula para esse trabalho.

E – módulo de elasticidade de Young (kN/m²)

A – área da seção transversal do elemento (m²)

L – comprimento longitudinal de cada elemento (m)

I – momento de inércia da seção transversal do elemento

[K_{ii}] – matriz 3 x 3 para rigidez de um nó i em i, sendo:

- rigidez na direção x-x (kN/m);
- rigidez na direção y-y (kN/m);
- rigidez na direção y-z (kN/rad);
- rigidez em torno de z-z (kN.m/rad);
- demais direções foram desconsideradas, pois possuem rigidez nula para esse trabalho.

[K_{ij}] – matriz 3 x 3 para rigidez de um nó i em j, sendo:

- rigidez na direção x-x (kN/m);
- rigidez na direção y-y (kN/m);
- rigidez na direção y-z (kN/rad);
- rigidez em torno de z-z (kN.m/rad);
- demais direções foram desconsideradas, pois possuem rigidez nula para esse trabalho.

[K_{ji}] – matriz 3 x 3 para rigidez de um nó j em i, sendo:

- rigidez na direção x-x (kN/m);
- rigidez na direção y-y (kN/m);
- rigidez na direção y-z (kN/rad);
- rigidez em torno de z-z (kN.m/rad);
- demais direções foram desconsideradas, pois possuem rigidez nula para esse trabalho.

[K_{jj}] – matriz 3 x 3 para rigidez de um nó j em j, sendo:

- a) rigidez na direção x-x (kN/m);
- b) rigidez na direção y-y (kN/m);
- c) rigidez na direção y-z (kN/rad);
- d) rigidez em torno de z-z (kN.m/rad);
- e) demais direções foram desconsideradas, pois possuem rigidez nula para esse trabalho.

$Q(x)$ – esforço cortante de uma seção da viga em x (kN)

$Q(g_i)$ – esforço cortante de uma seção da viga em um ponto inicial g_i do trecho (kN)

$R(g_i)$ – reação de apoio vertical no ponto inicial g_i do trecho (kN)

$V(g_i)$ – valor da carga pontual na direção y aplicada no ponto inicial g_i do trecho (kN)

q_r – valor da carga distribuída de valor constante ao longo do trecho (kN/m)

$q_t(g_i)$ – valor de carga distribuída triangular no ponto inicial g_i do trecho (kN/m)

$q_t(g_{i+1})$ – valor de carga distribuída triangular no ponto final g_{i+1} do trecho (kN/m)

x – valor da abscissa do ponto em que se deseja o valor do cortante, medido a partir da origem da viga, em g_0 (m)

e_0 – excentricidade inicial do cabo de protensão em relação ao centro de gravidade da seção da viga em sua ponta mais à esquerda (m)

e_i – excentricidade da armadura de protensão no ponto de inflexão das parábolas à esquerda de um trecho da viga (m)

e_j – excentricidade da armadura de protensão no ponto de inflexão das parábolas à direita de um trecho da viga (m)

CG – centro de gravidade geométrico da seção da viga (m)

y_{si} , y_{ii} , y_{ij} , y_{sj} – equação da excentricidade de cada segmento de parábola em relação ao CG, que descreve a geometria do cabo de protensão (m), sendo:

- a) s_i – relativo à parábola superior à esquerda de um trecho da viga
- b) ii – relativo à parábola inferior à esquerda de um trecho da viga

- c) f_{ij} – relativo à parábola inferior à direita de um trecho da viga
- d) f_{sj} – relativo à parábola superior à direita de um trecho da viga

$f_{si}, f_{ii}, f_{ij}, f_{jj}$ – flecha de cada segmento de parábola que descreve a geometria do cabo de protensão, que relaciona a diferença de y entre o cume e início desse segmento (m), sendo:

- a) f_{si} – relativo à parábola superior à esquerda de um trecho da viga
- b) f_{ii} – relativo à parábola inferior à esquerda de um trecho da viga
- c) f_{ij} – relativo à parábola inferior à direita de um trecho da viga
- d) f_{sj} – relativo à parábola superior à direita de um trecho da viga

$l_{si}, l_{ii}, l_{ij}, l_{jj}$ – comprimento de projeção de cada segmento de parábola que descreve a geometria do cabo de protensão no eixo das abscissas (m), sendo:

- a) l_{si} – relativo à parábola superior à esquerda de um trecho da viga
- b) l_{ii} – relativo à parábola inferior à esquerda de um trecho da viga
- c) l_{ij} – relativo à parábola inferior à direita de um trecho da viga
- d) l_{sj} – relativo à parábola superior à direita de um trecho da viga

$\Delta h_i, \Delta h_j$ – máxima diferença de cota entre as parábolas em um lado de um trecho da viga (m), sendo:

- a) i – relativo ao lado esquerdo do trecho;
- b) j – relativo ao lado direito do trecho

I_{CG} – momento de inércia de área no centro de gravidade da seção transversal da viga (m^4)

b – largura de uma seção transversal retangular da viga (m)

h – altura de uma seção transversal da viga (m)

ΣI_{xCGi} : somatório dos momentos de inércia de cada parte i de uma figura decomposta (triângulos e retângulos) com relação aos seus centros geométricos verticais locais (m^4)

$\Sigma A_i d_{yi}^2$: somatório do produto entre a área de cada parte i de uma figura decomposta (triângulos e retângulos) pelo quadrado da distância do centro geométrico de cada parte i em relação ao centro geométrico da seção (m^4)

SUMÁRIO

1 INTRODUÇÃO	17
2 DIRETRIZES DA PESQUISA.....	19
2.1 QUESTÃO DA PESQUISA.....	19
2.2 OBJETIVOS DA PESQUISA	19
2.2.1 Objetivo Principal	19
2.2.2 Objetivos Secundários.....	19
2.3 PRESSUPOSTOS.....	20
2.4 DELIMITAÇÕES.....	20
2.5 LIMITAÇÕES	21
2.6 DELINEAMENTO.....	21
3 ANÁLISE DA ESTRUTURA	25
3.1 MATRIZ DE RIGIDEZ DO TIPO PÓRTICO PLANO	25
3.2 MATRIZ DE RIGIDEZ GLOBAL	26
3.3 CONSIDERAÇÃO DAS RESTRIÇÕES	26
3.4 OBTENÇÃO DAS REAÇÕES	30
3.5 OBTENÇÃO DOS DIAGRAMAS DE ESFORÇOS.....	31
3.6 ESFORÇOS SOLICITANTES.....	32
4 DESCRIÇÃO DAS GEOMETRIAS.....	33
4.1 GEOMETRIA DA PROTENSÃO	33
4.1.1 Determinação das excentricidades intermediárias.....	35
4.1.2 Determinação das equações das parábolas	36
4.2 GEOMETRIA DA SEÇÃO DA VIGA	40
4.2.1 Do Cálculo das Propriedades	40
5 PROTENSÃO DO CONCRETO	42
5.1 TIPOS DE PROTENSÃO	42
5.2 MATERIAIS	44
5.2.1 Tensão do Concreto	44
5.2.2 Aço Para Armadura Passiva	48
5.2.3 Aço da Armadura Ativa.....	49
6 DETERMINAÇÃO DA PROTENSÃO.....	52

6.1 MÉTODO DO BALANCEAMENTO DE CARGAS.....	52
6.2 PROTENSÃO EQUIVALENTE EM CADA TRECHO	56
6.3 PROTENSÃO DA VIGA COMO UM TODO	58
7 VERIFICAÇÕES DA VIGA	59
7.1 TENSÕES ADMISSÍVEIS DO CONCRETO	59
8 PERDAS DA FORÇA DE PROTENSÃO.....	64
8.1 PERDAS IMEDIATAS	64
8.1.1 Perda Imediata por Encurtamento Elástico do Concreto	64
8.1.2 Perda Imediata por Atrito	69
8.1.3 Perdas Por Recuo de Ancoragem.....	72
8.2 PERDAS PROGRESSIVAS	76
8.2.1 Relaxação do Aço.....	76
8.2.2 Deformações do Concreto no Tempo.....	77
8.2.2.1 <i>Fluência do Concreto</i>	78
8.2.2.2 <i>Retração do Concreto</i>	82
9 IMPLEMENTAÇÃO COMPUTACIONAL.....	85
9.1 ELEMENTOS DO PROGRAMA	85
9.1.1 Geometria	87
9.1.1.1 <i>Seção</i>	87
9.1.1.2 <i>Viga</i>	88
9.1.1.3 <i>Grids</i>	88
9.1.2 Cargas	89
9.1.3 Restrições.....	91
9.1.4 Instantes temporais	93
9.1.5 Materiais.....	94
9.1.6 Resultados da Estrutura	95
9.2 ALGORITMOS DE CÁLCULO.....	96
9.2.1 Ambiente de Desenvolvimento	97
9.2.1.1 <i>Organização da Linguagem C#</i>	97
9.2.2 Implementação Da Seção Transversal da Viga	98
9.2.2.1 <i>Cálculo da Área</i>	100
9.2.2.2 <i>Cálculo do Centro Geométrico Vertical</i>	101
9.2.2.3 <i>Cálculo do Momento de Inércia de Área em Torno do Eixo Horizontal</i>	102
9.2.2.4 <i>Cálculo dos Módulos Resistentes à Flexão</i>	103
9.2.3 Implementação dos Carregamentos da Viga	103

9.2.4 Implementação das Restrições Nodais.....	104
9.2.5 Implementação dos Nós	105
9.2.6 Implementação dos Materiais	106
<i>9.2.6.1 Concreto</i>	<i>106</i>
9.2.7 Implementação do Método da Rigidez	109
<i>9.2.7.1 Implementação do Sistema Matricial.....</i>	<i>109</i>
9.2.8 Implementação das Solicitações	115
9.3 VALIDAÇÃO DOS RESULTADOS	118
10 CONSIDERAÇÕES FINAIS.....	122
REFERÊNCIAS	125
ANEXO A – MANUAL DO PROGRAMA.....	126

1 INTRODUÇÃO

Este trabalho propõe a criação de um programa de computador para projeto de vigas protendidas. Pretende-se o desenvolvimento incremental da codificação das tarefas de projeto, de modo que as realizações do software sejam sempre possíveis de expansão e melhoramento. Isso permite flexibilidade na implementação desse trabalho, de modo que, como objetivo inicial, será preparar e possibilitar a entrada dos dados necessários ao cálculo das vigas, bem como sua solução com a apresentação dos resultados das solicitações sobre elas. Com essa base será possível implementar futuramente um tipo de projeto completo de viga protendida, e, bem como, diversos tipos de vigas, protensões, e o que mais julgar-se necessário para tornar cada vez mais eficiente o trabalho dos engenheiros pelo uso de tal ferramenta.

Esse trabalho foi limitado a tratar apenas o caso de protensão por pós-tração, para uma viga individual, com tipos de seção pré-determinados, com a aplicação de diferentes condições de apoio a critério do usuário, bem como diferentes tipos de cargas de tipos também pré-determinados. O critério de escolha dos tipos de seções possíveis foi baseado nos tipos mais comumente encontrados em concretos pré-fabricados, sendo assim: formato retangular e formato em I.

Para resolução da estrutura foi adotado o método dos deslocamentos por análise matricial, com matriz de rigidez do tipo pórtico plano, sendo do tipo linear. Dessa forma as condições de contorno a esse tipo de estrutura determinaram também a seleção dos tipos de apoio possíveis. As condições de contorno serão aplicadas por meio de molas (rigidezes equivalentes dos vínculos da viga com a estrutura), aceitando-se essas restrições na direção vertical, horizontal e circular em cada nó. As cargas selecionadas para implementação basearam-se nas mais usuais utilizadas em projetos, sendo pontuais verticais ou de momento, e distribuídas constantes.

Os materiais utilizados são o concreto, para a viga, e aços para as armaduras passivas e ativas. Uma biblioteca de materiais deve permitir ao engenheiro a seleção desses materiais, bem como a personalização dos mesmos por meio de ajuste de parâmetros. A norma NBR 6118:2014 é aplicada pelo programa em todas as etapas do desenvolvimento de cada projeto por meio dele, indicando ao usuário os critérios necessários ao entendimento de erros ou problemas.

A programação desse trabalho utiliza linguagem C# e XAML, por meio do ambiente de programação Microsoft Visual Studio 2015 (MSVS2015), com uso do subsistema gráfico *Windows Presentation Foundation* (WPF) do .NET Framework 3.5. Foi escolhido o WPF pois esse permite grande versatilidade de programação, como também maior portabilidade. As bibliotecas *runtime* já se encontram inclusas em todas as versões do Microsoft Windows desde o Windows Vista, minimizando questões de instalação do aplicativo final.

2 DIRETRIZES DA PESQUISA

As diretrizes para desenvolvimento desse trabalho estão descritas nos itens a seguir.

2.1 QUESTÃO DA PESQUISA

A questão do trabalho é: que formulações são necessárias ao dimensionamento de uma viga de concreto com múltiplos vãos, para dimensionamento adequado às normas brasileiras, de uma armadura de protensão por pós-tração, de modo que seja possível sua implementação em algoritmos computacionais?

2.2 OBJETIVOS DA PESQUISA

Os objetivos da pesquisa são classificados em principal e secundários, descritos a seguir.

2.2.1 Objetivo Principal

O objetivo principal do trabalho consiste em definir os elementos necessários para implementação de um programa de computador capaz de dimensionar a armadura de protensão pós-traçada para vigas de concreto de múltiplos vãos, formada por segmentos de parábolas pelo método do balanceamento das cargas, a partir de parâmetros definidos pelo engenheiro.

2.2.2 Objetivos Secundários

Os objetivos secundários do trabalho são:

- a) exposição dos dados e elementos necessários a serem informados em um programa que permita a análise de vigas e dimensionamento de armadura de protensão;

- b) exposição da formulação da análise matricial de estruturas utilizada em programas computacionais, particularizada para a análise de vigas;
- c) modelagem e descrição matemática da geometria de um cabo de protensão formado por segmentos de parábolas para seu armazenamento em uma estrutura de dados pelo computador;
- d) exposição do método de balanceamento de cargas para transformar o problema dos esforços gerados pelos cabos de protensão em carregamentos, de modo que a mesma formulação da alínea b permita ao computador analisar a estrutura com a protensão;
- e) descrição matemática das verificações ligadas à protensão como perdas imediatas e perdas progressivas;
- f) apresentação da implementação do código – diagramas que representem o funcionamento do código do programa implementado;
- g) apresentação de interface gráfica interativa, nos padrões dos *softwares* mais atuais, de modo a facilitar sua usabilidade;
- h) implementação de código em linguagem de programação bastante difundida em ambiente versátil que permita melhor uso dos recursos computacionais como uso do *mouse*, gráficos 3D, e portabilidade futura para outros tipos de dispositivos e sistemas, além do computador pessoal, como *smartphones* e *tablets*.

2.3 PRESSUPOSTOS

O trabalho tem por pressuposto a validade as especificações das normas brasileiras, em especial, a NBR 6118:2014 – Projeto de Estruturas de Concreto - Procedimento, como adequada aos objetivos desse trabalho, o método dos deslocamentos com análise matricial das estruturas como apropriado para cálculo das reações e esforços da viga, bem como os resultados produzidos pelo método de balanceamento de cargas para dimensionamento da armadura de protensão em vigas hiperestáticas, em que a curvatura do cabo de protensão é muito pequena.

2.4 DELIMITAÇÕES

O trabalho delimita-se ao estudo das formulações matemáticas necessárias ao método dos deslocamentos para análise matricial da estrutura a ao método do balanceamento de cargas para aplicação em protensão por pós-tração de vigas de concreto, e posterior implementação computacional. Esses modelos têm como particularidade:

- a) apenas estruturas de vigas de concreto de Cimento Portland;
- b) protensão realizada apenas por pós-tração da armadura ativa;
- c) geometria do cabo de protensão formada por segmentos de parábolas;
- d) implementação parcial do programa demonstrando seus principais mecanismos para solução do problema, não exaustivo em todas as etapas do projeto demonstrando sua capacidade para continuidade de um desenvolvimento futuro;
- e) o programa implementado será capaz de realizar a entrada de dados e o cálculo das solicitações, não sendo implementada a parte da protensão nesse trabalho.

2.5 LIMITAÇÕES

São limitações do trabalho:

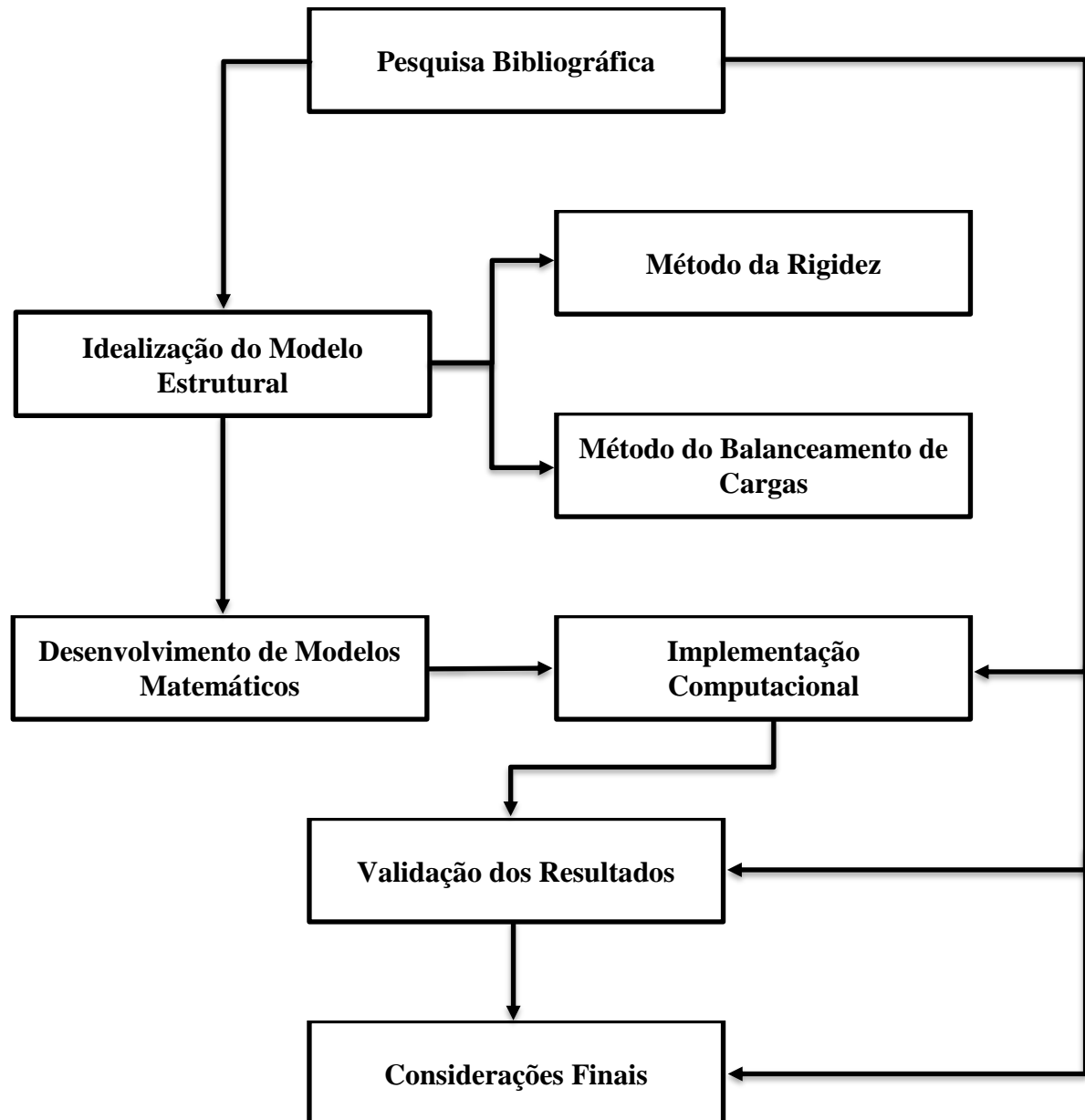
- a) as curvaturas nos cabos de protensão devem ser pequenas;
- b) os vãos da viga devem ser ter comprimento mínimo igual ao dobro da sua altura;
- c) a análise das estruturas é linear;
- d) são considerados apenas carregamentos estáticos;
- e) número de segmentos de parábolas para vãos igual a quatro e para balanços igual a dois.

2.6 DELINEAMENTO

O trabalho foi realizado a partir das etapas apresentadas a seguir, representadas na Figura 1, e descritas nos próximos parágrafos:

- a) pesquisa bibliográfica;
- b) idealização do modelo estrutural;
- c) desenvolvimento de modelos matemáticos;
- d) implementação computacional;
- e) validação dos resultados;
- f) considerações finais.

Figura 1 – Diagrama das etapas de pesquisa.



(fonte: elaborada pelo autor.)

A **pesquisa bibliográfica** está relacionada com todas as etapas do desenvolvimento do trabalho. O principal foco da pesquisa buscou aprofundar os conhecimentos na área do método da rigidez

para a resolução dos esforços da estrutura, no método do balanceamento de cargas para consideração dos esforços produzidos pela armadura de protensão sobre a viga, remetendo-o de volta ao método da rigidez, e sobre a norma brasileira no que tange as considerações adicionais que devem ser realizadas sobre armaduras de protensão por pós-tração bem como do comportamento dos materiais envolvidos, isto é, o concreto e o aço. Com esse aprofundamento foi possível relacionar todos os elementos envolvidos no problema para possibilitar a formulação de sua resolução. Por fim, foram aprofundados também os conhecimentos sobre linguagens de programação e de estrutura de dados para iniciar a implementação computacional dos dados do problema.

A **idealização do modelo estrutural** consiste basicamente em levantar todas as variáveis necessárias de um problema real e então idealizá-las em parâmetros que descrevam seu comportamento em um modelo de cálculo com razoável proximidade. Essa idealização envolve tanto a questão da estrutura da viga com seus carregamentos e restrições de deslocamentos, como sua geometria e o material que a compõe, quanto a questão dos efeitos da armadura de protensão com sua geometria e material, resultando na forma de um carregamento adicional sobre a viga.

No **desenvolvimento de modelos matemáticos** tem-se a apresentação da formulação matricial utilizada para modelagem da viga por elementos de pórticos planos pelo **método da rigidez** e suas considerações levando em conta a idealização do modelo estrutural, relacionando os parâmetros levantados anteriormente. Ainda nessa mesma fase também é apresentada a formulação do **método do balanceamento de cargas**, para através também dos parâmetros idealizados no modelo estrutural, chegar-se a como estes serão considerados matematicamente para transformação da armadura ativa em questão em carregamentos bem como o dimensionamento das excentricidades de suas parábolas de modo autônomo. Nessa fase foram adicionalmente levantadas as formulações da NBR 6118:2014 para considerações adicionais acerca do dimensionamento da protensão, como perdas imediatas e progressivas do aço de protensão e do concreto para possibilitar futuramente a continuidade do programa desenvolvido nesse trabalho.

A **implementação computacional** envolve não apenas a tradução dos modelos matemáticos desenvolvidos em linguagem de programação como também o desenvolvimento de uma interface com a qual o engenheiro irá interagir para poder fornecer os parâmetros necessários

de projeto, levantados na idealização do modelo. Essa fase corresponde ao desenvolvimento de rotinas e algoritmos de computador para executar os cálculos matemáticos definidos como, a partir dos parâmetros fornecidos para a viga, aplicando o método da rigidez, produzir os resultados de reações e solicitações sobre a viga. Devido à complexidade e tamanho dessas rotinas, grande parte será apresentada de forma esquemática com descrição detalhada para os modelos de cálculo implementados.

A **validação dos resultados** corresponde a produzir resultados da implementação computacional realizada e compará-los com os mesmos resultados obtidos por outros programas confiáveis, quando existirem. No caso do método da rigidez, o programa ftool foi adotado como uma boa referência.

As **considerações finais** buscam apresentar conclusões sobre o desenvolvimento do trabalho e sobre os resultados obtidos.

3 ANÁLISE DA ESTRUTURA

O tipo de análise realizado é linear, matricial, sendo empregado o método dos deslocamentos para resolução da estrutura. Em vista de limitar-se o trabalho a apenas uma viga nesse momento, os elementos componentes dessa matriz são do tipo pórtico plano; isto é, possuem coeficientes de rigidez dentro da matriz apenas para deslocamentos horizontais, verticais e giros em cada um de seus nós, definidos pelo *grid*. Ainda, pela opção de uma estrutura linear horizontal, não será necessário rotacionar nenhum dos elementos para montar a matriz de rigidez global da viga, pois todos já estão em posição compatível com um único referencial de eixos globais.

Sob estas condições, pode-se aplicar a Lei de Hooke, em sua forma matricial:

$$\{F\} = [K].\{U\} \quad (1)$$

A matriz de forças pode ser obtida a partir das cargas inseridas no programa pelo projetista. K será calculada com base nos materiais, geometria informada. A matriz de deslocamentos, inicialmente, será a parte incógnita da equação a ser resolvida. Para que seja possível a resolução da equação (1), condições de contorno são aplicadas com base nas rigidezes das restrições informadas, somando-as nas suas respectivas posições na matriz global, e igualando sua posição no vetor de forças a zero.

3.1 MATRIZ DE RIGIDEZ DO TIPO PÓRTICO PLANO

Para geração da matriz de rigidez de cada elemento local (K^L) é necessário o módulo de Yong (E) do concreto em questão, fornecido pela escolha de materiais no início de cada projeto dentro do *software*, o momento de inércia de área de cada seção (I), a área de cada seção (A), e o comprimento de cada trecho (L). Visto que a seção é constante, bem como o material, ao longo de toda a viga, a matriz de rigidez varia apenas com o comprimento do trecho obtido a partir do *grid* definido pelo usuário. Assim, pode-se escrever a matriz K como em GAVIN (2014):

$$K^L = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \quad (2)$$

Cada sub-matriz 3 x 3 (três por três) está relacionado a um nó no elemento. Sendo cada elemento composto por dois nós, e nomeando-os de i e j, podemos representar cada sub-matriz por um índice, conforme segue:

$$K^L = \begin{bmatrix} K_{ii} & K_{ij} \\ K_{ji} & K_{jj} \end{bmatrix} \quad (3)$$

3.2 MATRIZ DE RIGIDEZ GLOBAL

A matriz de rigidez global (K^G) é composta por super-nós. Cada super-nó é composto pela soma das rigidezes individuais de matrizes locais (K_{ii} , K_{ij} , K_{ji} , K_{jj}), em que os índices para essas rigidezes coincidam no super-nó da sobreposição. A equação (4) abaixo representa os três nós de uma viga, e a equação (5) a sua composição em uma matriz global:

$$K^{12} = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \quad K^{23} = \begin{bmatrix} K_{22} & K_{23} \\ K_{32} & K_{33} \end{bmatrix} \quad K^{13} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (4)$$

$$K^G = \begin{bmatrix} K_{11}^{12} + 0 & K_{12}^{12} & 0 \\ K_{21}^{12} & K_{22}^{12} + K_{22}^{23} & K_{23}^{23} \\ 0 & K_{32}^{23} & K_{33}^{23} + 0 \end{bmatrix} \quad (5)$$

3.3 CONSIDERAÇÃO DAS RESTRIÇÕES

A resolução do sistema matricial ocorre em etapas, de modo que inicialmente gera-se um sistema simplificado, para a redução do número de incógnitas, possibilitando assim sua solução.

Para cada rigidez inserida nos nós da viga, por meio das restrições aplicadas pelo usuário na coleta de dados, faz-se a soma desses valores em elementos específicos da matriz, zerando-se a força ou reação existente no vetor F naquela linha. A igualdade mantém-se, dessa forma, como pode ser demonstrado abaixo:

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} & K_{15} & K_{16} \\ K_{21} & K_{22} & K_{23} & K_{24} & K_{25} & K_{26} \\ K_{31} & K_{32} & K_{33} & K_{34} & K_{35} & K_{36} \\ K_{41} & K_{42} & K_{43} & K_{44} & K_{45} & K_{46} \\ K_{51} & K_{52} & K_{53} & K_{54} & K_{55} & K_{56} \\ K_{61} & K_{62} & K_{63} & K_{64} & K_{65} & K_{66} \end{bmatrix} \cdot \begin{Bmatrix} u_1 \\ v_1 \\ \varphi_1 \\ u_2 \\ v_2 \\ \varphi_2 \end{Bmatrix} = \begin{Bmatrix} H_1 \\ V_1 \\ M_{z1} \\ H_2 \\ V_2 \\ M_{z2} \end{Bmatrix} \quad (6)$$

$$K_{21} \cdot u_1 + K_{22} \cdot v_1 + K_{23} \cdot \varphi_1 + K_{24} \cdot u_2 + K_{25} \cdot v_2 + K_{26} \cdot \varphi_2 = V_1 \quad (7)$$

Havendo sido inserida uma rigidez vertical no nó 1, tem-se a equação:

$$k_{v1} \cdot v_1 = V_{R1} \quad (8)$$

$$V_{R1} = -V_1 \quad (9)$$

A reação na mola VR1 é oposta à V1 do vetor de forças, de modo obtém-se a equação abaixo:

$$V_1 = -k_{v1} \cdot v_1 \quad (10)$$

Igualando as duas equações (7) e (10), obtém-se:

$$K_{21} \cdot u_1 + K_{22} \cdot v_1 + K_{23} \cdot \varphi_1 + K_{24} \cdot u_2 + K_{25} \cdot v_2 + K_{26} \cdot \varphi_2 = -k_{v1} \cdot v_1 \quad (11)$$

$$K_{21} \cdot u_1 + (K_{22} + k_{v1}) \cdot v_1 + K_{23} \cdot \varphi_1 + K_{24} \cdot u_2 + K_{25} \cdot v_2 + K_{26} \cdot \varphi_2 = 0 \quad (12)$$

Observa-se aqui que a rigidez deve ser isotrópica, isto é, tanto compressão como tração oferecem mesma resistência, de modo que o sinal entre as rigidezes deverá ser de soma, independente da direção de força. Reescrevendo novamente na forma matricial, tem-se:

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} & K_{15} & K_{16} \\ K_{21} & K_{22} + k_{v1} & K_{23} & K_{24} & K_{25} & K_{26} \\ K_{31} & K_{32} & K_{33} & K_{34} & K_{35} & K_{36} \\ K_{41} & K_{42} & K_{43} & K_{44} & K_{45} & K_{46} \\ K_{51} & K_{52} & K_{53} & K_{54} & K_{55} & K_{56} \\ K_{61} & K_{62} & K_{63} & K_{64} & K_{65} & K_{66} \end{bmatrix} \cdot \begin{Bmatrix} u_1 \\ v_1 \\ \varphi_1 \\ u_2 \\ v_2 \\ \varphi_2 \end{Bmatrix} = \begin{Bmatrix} H_1 \\ 0 \\ M_{z1} \\ H_2 \\ V_2 \\ M_{z2} \end{Bmatrix} \quad (13)$$

Quando as restrições inseridas no programa forem vínculos perfeitos, de imediato sabe-se que o deslocamento naquela direção será nulo, de modo que outra simplificação é possível. No vetor de deslocamentos a posição correspondente ao vínculo perfeito deve ser zerada, bem como a posição equivalente no vetor de forças. Segue demonstração:

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} & K_{15} & K_{16} \\ K_{21} & K_{22} & K_{23} & K_{24} & K_{25} & K_{26} \\ K_{31} & K_{32} & K_{33} & K_{34} & K_{35} & K_{36} \\ K_{41} & K_{42} & K_{43} & K_{44} & K_{45} & K_{46} \\ K_{51} & K_{52} & K_{53} & K_{54} & K_{55} & K_{56} \\ K_{61} & K_{62} & K_{63} & K_{64} & K_{65} & K_{66} \end{bmatrix} \cdot \begin{Bmatrix} u_1 \\ v_1 \\ \varphi_1 \\ u_2 \\ v_2 \\ \varphi_2 \end{Bmatrix} = \begin{Bmatrix} H_1 \\ V_1 \\ M_{z1} \\ H_2 \\ V_2 \\ M_{z2} \end{Bmatrix} \quad (14)$$

Havendo sido inserido um vínculo perfeito restringindo o giro no nó 1, tem-se a equação:

$$k_{\varphi 1} \cdot \varphi_1 = M_{z1} = k_{\varphi 1} \cdot 0 = M_{z1} \quad (15)$$

Para satisfazer a igualdade, M_{z1} deve ser necessariamente zero ou ocorreria um giro no ponto.

Reescrevendo-se o sistema, tem-se:

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} & K_{15} & K_{16} \\ K_{21} & K_{22} & K_{23} & K_{24} & K_{25} & K_{26} \\ K_{31} & K_{32} & K_{33} & K_{34} & K_{35} & K_{36} \\ K_{41} & K_{42} & K_{43} & K_{44} & K_{45} & K_{46} \\ K_{51} & K_{52} & K_{53} & K_{54} & K_{55} & K_{56} \\ K_{61} & K_{62} & K_{63} & K_{64} & K_{65} & K_{66} \end{bmatrix} \cdot \begin{Bmatrix} u_1 \\ v_1 \\ 0 \\ u_2 \\ v_2 \\ \varphi_2 \end{Bmatrix} = \begin{Bmatrix} H_1 \\ V_1 \\ 0 \\ H_2 \\ V_2 \\ M_{z2} \end{Bmatrix} \quad (16)$$

A linha em que ocorre essa eliminação acaba por não ter incógnitas, e então pode ser eliminada do cálculo. Computacionalmente é mais simples reescrever a matriz, preenchendo a coluna e linha correspondente com zeros e uns, ao invés de alterar-se as dimensões da matriz, ficando dessa forma o seguinte resultado:

$$\begin{bmatrix} K_{11} & K_{12} & 0 & K_{14} & K_{15} & K_{16} \\ K_{21} & K_{22} & 0 & K_{24} & K_{25} & K_{26} \\ 0 & 0 & 1 & 0 & 0 & 0 \\ K_{41} & K_{42} & 0 & K_{44} & K_{45} & K_{46} \\ K_{51} & K_{52} & 0 & K_{54} & K_{55} & K_{56} \\ K_{61} & K_{62} & 0 & K_{64} & K_{65} & K_{66} \end{bmatrix} \cdot \begin{Bmatrix} u_1 \\ v_1 \\ 0 \\ u_2 \\ v_2 \\ \varphi_2 \end{Bmatrix} = \begin{Bmatrix} H_1 \\ V_1 \\ 0 \\ H_2 \\ V_2 \\ M_{z2} \end{Bmatrix} \quad (17)$$

Outro caso de simplificação possível é quando existem deslocamentos pré-determinados em um nó da estrutura. Para esses casos, o valor da incógnita (deslocamento) na linha em questão é sabido, podendo-se reescrever o sistema para eliminar essa linha que pode ter alguma incógnita no vetor de forças, realizando as seguintes operações:

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} & K_{15} & K_{16} \\ K_{21} & K_{22} & K_{23} & K_{24} & K_{25} & K_{26} \\ K_{31} & K_{32} & K_{33} & K_{34} & K_{35} & K_{36} \\ K_{41} & K_{42} & K_{43} & K_{44} & K_{45} & K_{46} \\ K_{51} & K_{52} & K_{53} & K_{54} & K_{55} & K_{56} \\ K_{61} & K_{62} & K_{63} & K_{64} & K_{65} & K_{66} \end{bmatrix} \cdot \begin{Bmatrix} u_1 \\ v_1 \\ \varphi_1 \\ u_2 \\ v_2 \\ \varphi_2 \end{Bmatrix} = \begin{Bmatrix} H_1 \\ V_1 \\ M_{z1} \\ H_2 \\ V_2 \\ M_{z2} \end{Bmatrix} \quad (18)$$

Aplicando-se um deslocamento vertical fixo no nó 2, δ_2 , tem-se:

$$v_2 = \delta_2 \therefore V_2 = 0 \quad (19)$$

Como δ_2 no caso seria o resultado do deslocamento final, a estrutura estaria em equilíbrio para esse valor, podendo-se anular a força correspondente do vetor de forças. Ainda, com a imposição desse deslocamento nesse nó, as demais equações do sistema podem eliminar essa incógnita, trocando-a para o lado das forças conforme segue:

$$K_{11} \cdot u_1 + K_{12} \cdot v_1 + K_{13} \cdot \varphi_1 + K_{14} \cdot u_2 + K_{15} \cdot \delta_2 + K_{16} \cdot \varphi_2 = H_1 \quad (20)$$

$$K_{11} \cdot u_1 + K_{12} \cdot v_1 + K_{13} \cdot \varphi_1 + K_{14} \cdot u_2 + K_{16} \cdot \varphi_2 = H_1 - K_{15} \cdot \delta_2 \quad (21)$$

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} & 0 & K_{16} \\ K_{21} & K_{22} & K_{23} & K_{24} & K_{25} & K_{26} \\ K_{31} & K_{32} & K_{33} & K_{34} & K_{35} & K_{36} \\ K_{41} & K_{42} & K_{43} & K_{44} & K_{45} & K_{46} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ K_{61} & K_{62} & K_{63} & K_{64} & K_{65} & K_{66} \end{bmatrix} \cdot \begin{Bmatrix} u_1 \\ v_1 \\ \varphi_1 \\ u_2 \\ \delta_2 \\ \varphi_2 \end{Bmatrix} = \begin{Bmatrix} H_1 - K_{15} \cdot \delta_2 \\ V_1 \\ M_{z1} \\ H_2 \\ 0 \\ M_{z2} \end{Bmatrix} \quad (22)$$

Esse procedimento pode ser repetido para cada linha da matriz, até que por fim tem-se uma simplificação do sistema computacional reduzido de uma dimensão.

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} & 0 & K_{16} \\ K_{21} & K_{22} & K_{23} & K_{24} & 0 & K_{26} \\ K_{31} & K_{32} & K_{33} & K_{34} & 0 & K_{36} \\ K_{41} & K_{42} & K_{43} & K_{44} & 0 & K_{46} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ K_{61} & K_{62} & K_{63} & K_{64} & 0 & K_{66} \end{bmatrix} \cdot \begin{Bmatrix} u_1 \\ v_1 \\ \varphi_1 \\ u_2 \\ \delta_2 \\ \varphi_2 \end{Bmatrix} = \begin{Bmatrix} H_1 - K_{15} \cdot \delta_2 \\ V_1 - K_{25} \cdot \delta_2 \\ M_{z1} - K_{35} \cdot \delta_2 \\ H_2 - K_{45} \cdot \delta_2 \\ 0 \\ M_{z2} - K_{65} \cdot \delta_2 \end{Bmatrix} \quad (23)$$

Por fim, para que a estrutura esteja estável, caso não existe restrição horizontal na estrutura, o programa considera o deslocamento horizontal do primeiro nó igual a zero para possibilitar a resolução da matriz, sem interferir nos resultados. Assim, de forma genérica, seguindo as demonstrações anteriores, o sistema final a ser resolvido seria:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & K_{22} + k_{v1} & 0 & K_{24} & 0 & K_{26} \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & K_{42} & 0 & K_{44} & 0 & K_{46} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & K_{62} & 0 & K_{64} & 0 & K_{66} \end{bmatrix} \cdot \begin{Bmatrix} 0 \\ v_1 \\ 0 \\ u_2 \\ \delta_2 \\ \varphi_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ -K_{25} \cdot \delta_2 \\ 0 \\ H_2 - K_{45} \cdot \delta_2 \\ 0 \\ M_{z2} - K_{65} \cdot \delta_2 \end{Bmatrix} \quad (24)$$

As incógnitas finais seriam v_1 , u_2 , φ_2 , com carregamentos existentes ou não H_1 , H_2 e M_{z2} , de modo que o sistema é determinado.

3.4 OBTENÇÃO DAS REAÇÕES

As reações sobre a estrutura são determinadas pelo uso dos deslocamentos obtidos com a resolução do sistema $\{K\} \cdot [U] = \{F\}$, com diversos passos de simplificação sobre o sistema. Com o resultado de U obtido desse sistema, pode-se reuplicar seus valores no sistema original, obtendo-se o vetor F . Com o resultado de F , subtraem-se então as forças aplicadas sobre os nós, resultando nas reações para cada um dos mesmos.

$$\begin{Bmatrix} H_1 \\ V_1 \\ M_{z1} \\ H_2 \\ V_2 \\ M_{z2} \end{Bmatrix} = \begin{Bmatrix} R_{H1} \\ R_{V1} \\ R_{Mz1} \\ R_{H2} \\ R_{V2} \\ R_{Mz2} \end{Bmatrix} + \begin{Bmatrix} F_{H1} \\ F_{V1} \\ F_{Mz1} \\ F_{H2} \\ F_{V2} \\ F_{Mz2} \end{Bmatrix} \quad \therefore \begin{Bmatrix} R_{H1} \\ R_{V1} \\ R_{Mz1} \\ R_{H2} \\ R_{V2} \\ R_{Mz2} \end{Bmatrix} = \begin{Bmatrix} H_1 - F_{H1} \\ V_1 - F_{V1} \\ M_{z1} - F_{Mz1} \\ H_2 - F_{H2} \\ V_2 - F_{V2} \\ M_{z2} - F_{Mz2} \end{Bmatrix} \quad (25)$$

3.5 OBTENÇÃO DOS DIAGRAMAS DE ESFORÇOS

A combinação das reações calculadas pela análise matricial nos nós criados ao longo da viga, combinado com os carregamentos de forma contínua permite a obtenção dos diagramas de esforços para qualquer ponto da barra. Computacionalmente pode-se dividir os trechos em funções conforme pontos de mudança de carregamentos ou reações, e criar um conjunto de funções válidas para cada trecho entre grids que resultarão em cada um dos esforços – normal, cortante ou de momento de flexão, no caso desse trabalho.

Sendo um *grid* existente em n pontos, nomeia-se cada um de g_0 até g_{n-1} , as reações e cargas pontuais poderão estar somente em algum ponto g_i , e as cargas distribuídas de g_i até g_{i+1} . Dessa forma, pode-se gerar um sistema de equações de forma analítica que forneça a sollicitação para qualquer coordenada ao longo da viga. Através de um eixo de coordenadas, chamado de eixo x com origem em g_0 , teremos funções $N(x)$, $Q(x)$ e $M(x)$ (para normal, cortante e momento de flexão respectivamente), entre cada trecho $g_i g_{i+1}$. Os esforços normais serão provenientes da protensão apenas, de modo que esse ponto será discutido mais adiante, bem como a contribuição do momento de flexão por parte da armadura ativa. Assim, expressões de cortante serão dadas por:

$$Q(x)_{g_i g_{i+1}} = Q(g_i) + R(g_i) - V(g_i) - q_r \cdot (x - g_i) \quad (26)$$

Onde,

$Q(x)_{g_i g_{i+1}}$ – valor do esforço cortante em função de x entre pontos do grid g_i e g_{i+1} ;

x – posição da abscissa ao longo da peça;

$Q(g_i)$ – valor do esforço cortante no ponto do *grid* g_i ;

$R(g_i)$ – valor da reação, caso existente, no *grid* g_i ;

$V(g_i)$ – carga pontual aplicada no *grid* g_i ;

q_r – valor taxa de carga constante no trecho em questão;

g_i ou g_{i+1} – valor da abscissa da posição do *grid* i ou $i+1$.

Cargas pontuais e reações irão gerar um degrau no gráfico e as distribuídas constantes uma reta inclinada com a declividade no valor de q_r .

Para o momento de flexão, a lógica aplicada é semelhante à usada para o cortante.

3.6 ESFORÇOS SOLICITANTES

Os esforços solicitantes da viga podem variar em função do tempo e possuir determinada duração, de modo que estes influenciam no resultado e desempenho final da protensão, pois esta última possui fatores de perda dependentes do tempo e desses esforços sofridos. Desse modo, o programa desenvolvido para esse trabalho associa a cada entrada de carregamento informações cronológicas sobre sua atuação e seu tipo (variável ou permanente) para posteriormente poder realizar descontos de perdas da tensão que a armadura irá sofrer. Além das perdas da armadura, também são computados os esforços sofridos pela viga a cada instante para se comparar com as resistências dos materiais, de modo a garantir a segurança da peça durante toda sua vida útil.

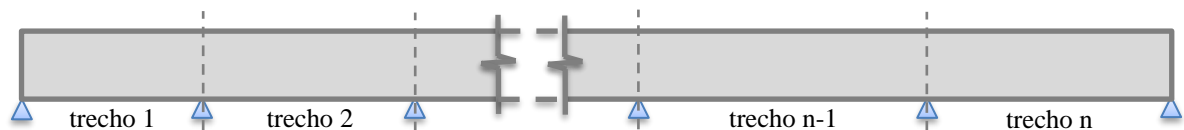
4 DESCRIÇÃO DAS GEOMETRIAS

Cada elemento geométrico dentro do programa será formulado de maneira paramétrica, a partir de um mínimo de dados necessários a serem fornecidos pelo usuário, sendo o restante deduzido pelos algoritmos a serem desenvolvidos.

4.1 GEOMETRIA DA PROTENSÃO

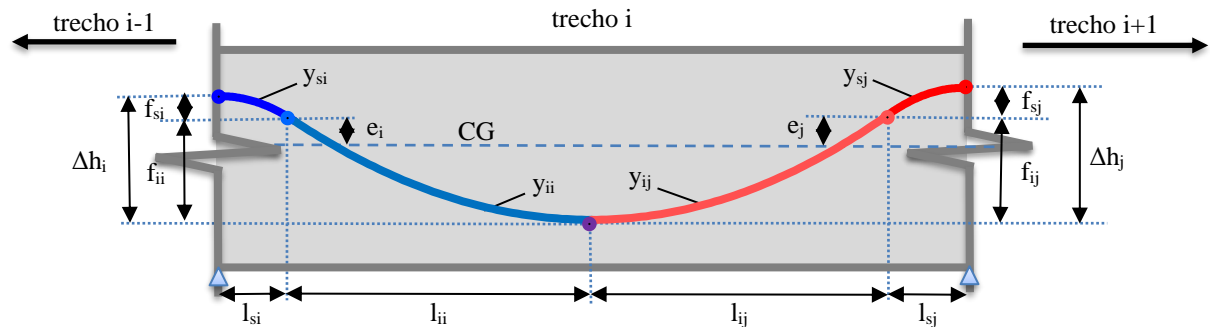
A geometria da protensão ao longo da viga será composta de trechos de parábolas, parametrizadas em função de uma excentricidade (em relação ao centro de gravidade da seção transversal), da diferença de cota entre apoio e o fim do trecho, e do comprimento sobre o qual cada parábola se estende. Cada vão entre dois apoios consecutivos é formado por quatro segmentos de parábolas, sendo que para os apoios externos uma parábola terá comprimento zero, equivalendo a uma composição de apenas três parábolas. A Figura 3 a seguir demonstra como essa segmentação da geometria do cabo é feita em função de segmentos de parábolas por trecho de viga, composta por n trechos (Figura 2).

Figura 2 – Segmentação da viga em trechos.



(fonte: elaborada pelo autor.)

Figura 3 – Segmentação da armadura de protensão em trechos de parábolas.



(fonte: elaborada pelo autor.)

Para cada trecho considerado existem 4 (quatro) parábolas que descrevem a geometria da protensão no concreto, sendo que no caso de trechos de extremidade da viga, a parábola mais à extremidade possuirá comprimento 0 (zero), equivalendo a um trecho de apenas três parábolas. As curvas superiores do trecho (ou nas extremidades do mesmo) e seus parâmetros receberam o sub-índice “s”, e as inferiores de maneira semelhante o sub-índice “i”. Ainda um segundo sub-índice foi utilizado para melhor identificar cada parábola individualmente, sendo “i” utilizado para os dois segmentos mais à esquerda do trecho, e “j” para os dois segmentos mais à direita.

As seguintes considerações foram feitas para definição das funções “y” que irão descrever cada uma das parábolas do cabo ao longo de toda a viga:

- todas as parábolas possuem continuidade entre si, isto é, a derivada no ponto de encontro entre parábolas é idêntico para ambas funções;
- os parâmetros “f” representam a flecha de cada parábola, sendo medida entre a maior e a menor cota (y) para cada segmento (sempre positiva);
- o parâmetro “e” representa a excentricidade das duas parábolas à esquerda ou à direita do trecho em relação ao centro de gravidade da viga;
- as funções “y” são referenciadas a partir do centro de gravidade “CG” da viga, sendo positiva para baixo e negativa para cima deste;
- o parâmetro “Δh” representa a flecha total entre as duas curvas à esquerda ou à direita do trecho, ou seja, $f_i + f_s$ para um dos lados do trecho;
- a letra “l” representa o comprimento do início ao fim de cada parábola individual;

- g) para trechos extremos da viga, simplesmente apoiados, a parábola superior do lado externo não existe, mantendo-se apenas as parábolas inferiores e uma parábola superior no lado que possui continuidade da viga;
- h) para vigas bi apoiadas, existem somente as parábolas inferiores;
- i) para trechos em balanço, existem apenas as duas parábolas referentes ao nó do lado em que a viga possui continuidade ou engastamento.

Os parâmetros que são variáveis, isto é, que dependem de valor informado pelo usuário no programa são o comprimento de cada parábola (l), a variação de altura (Δh), e uma excentricidade inicial (e_0) à esquerda do primeiro trecho da viga. Todos os demais parâmetros serão calculados em função dessas três entradas, um único e_0 , dois Δh para cada trecho, e quatro l 's para cada trecho.

4.1.1 Determinação das excentricidades intermediárias

A excentricidade dos demais trechos pode apenas ser calculada com base em e_0 , caso contrário as parábolas sofreriam uma descontinuidade entre trechos, ou o usuário teria de ajustar manualmente Δh 's e l 's específicos para compatibilizar as funções. Para manter a continuidade deduz-se as seguintes fórmulas para e_i e e_j :

$$f_{ii} - e_i = f_{ij} + e_j \quad (1)$$

(no meio do trecho para y_{ii} e y_{ij})

$$f_{sj} - e_j = f_{si} + e_i$$

$$\left(\begin{array}{l} \text{para } y_{sj} \text{ do trecho } i - 1 \text{ e } y_{si} \text{ do trecho } i, \\ \text{ou, para } y_{sj} \text{ do trecho } i \text{ e } y_{si} \text{ do trecho } i + 1 \end{array} \right) \quad (2)$$

Assim obtém-se os valores de maneira incremental, iniciando pelo primeiro trecho em que e_i será o próprio e_0 , conforme apresentado na Tabela 1, de maneira genérica:

Tabela 1 – Fórmulas para determinação das excentricidades de cada parábola.

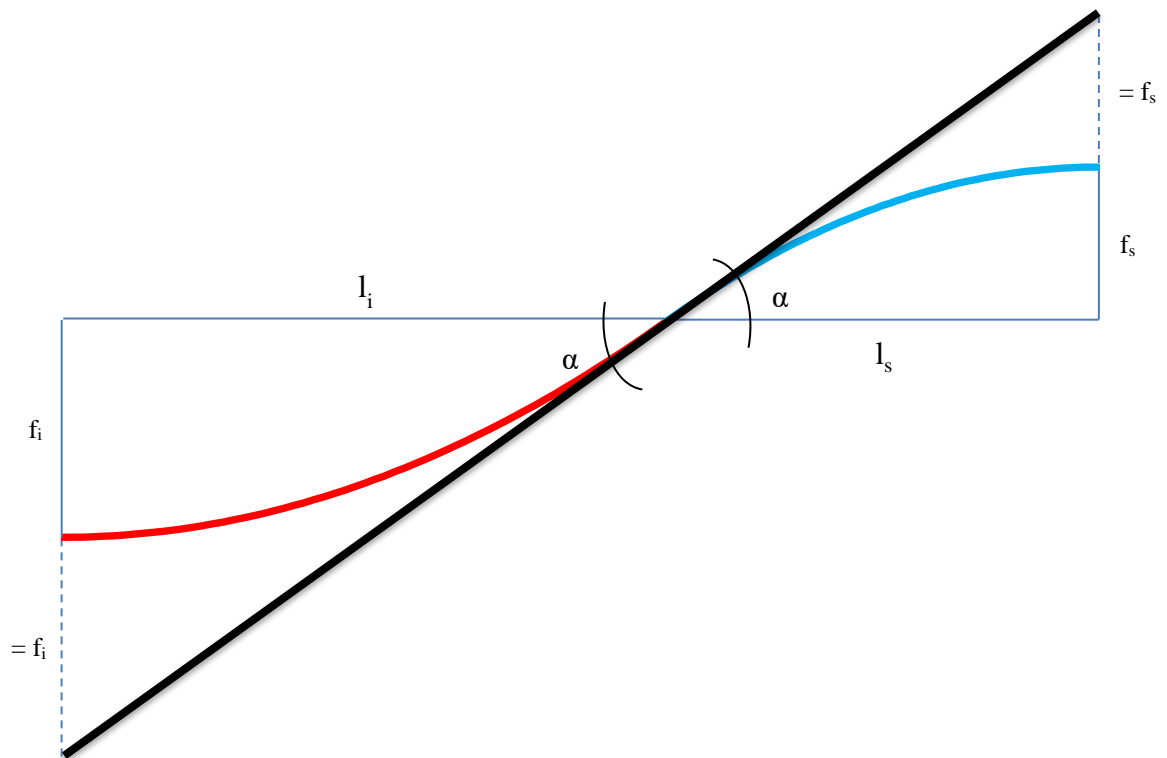
trecho	parâmetro	função
1	e_{i1}	$=e_0$
	e_{j1}	$=e_{i1}+f_{ii1}-f_{ij1}$
2	e_{i2}	$=e_{j1}+f_{si2}-f_{sj1}$
	e_{j2}	$=e_{i2}+f_{ii2}-f_{ij2}$
3	e_{i3}	$=e_{j2}+f_{si3}-f_{sj2}$
	e_{j3}	$=e_{i3}+f_{ii3}-f_{ij3}$
...	...	
n-1	$e_{i(n-1)}$	$=e_{j(n-2)}+f_{si(n-1)}-f_{sj(n-2)}$
	$e_{j(n-1)}$	$=e_{i(n-1)}+f_{ii(n-1)}-f_{ij(n-1)}$
n	e_{in}	$=e_{j(n-1)}+f_{sin}-f_{sj(n-1)}$
	e_{jn}	$=e_{in}+f_{iin}-f_{ijn}$

(fonte: elaborada pelo autor.)

4.1.2 Determinação das equações das parábolas

Todas as parábolas possuem uma das extremidades com declividade zero, isto é, derivada nula, e um outro ponto com um certo ângulo. Com base no Δh e l para cada grupo de duas parábolas (as duas mais à esquerda de cada trecho ou as duas mais à direita de cada trecho), é possível determinar o valor de f para cada uma delas, conforme segue demonstração.

Figura 4 – Relação de flechas e ângulo do ponto de inflexão entre parábolas.



(fonte: elaborada pelo autor.)

Tomando-se x e y iguais a zero no encontro das parábolas, em vermelha e em azul da Figura 4, e crescente para baixo, e sabendo que o ângulo α para ambas deve ser o mesmo como forma de garantir a continuidade entre as curvas, pode-se demonstrar que a relação de flechas segue as proporções de dois triângulos retângulos circunscritos às parábolas, com altura igual a dois f , e base l .

Para a cura em vermelho da Figura 4 determina-se:

$$y_i(x) = a_i x^2 + b_i x + c_i \quad (3)$$

$$y_i(0) = a_i \cdot 0 + b_i \cdot 0 + c_i = 0 \therefore c_i = 0 \quad (4)$$

$$y_i(-l_i) = f_i = a_i l_i^2 - b_i l_i \quad (5)$$

$$y'_i(x) = 2a_ix + b_i \quad (6)$$

$$y'_i(-l_i) = 0 = -2a_il_i + b_i \therefore b_i = 2a_il_i \quad (7)$$

$$f_i = a_il_i^2 - 2a_il_i^2 \therefore a_i = -\frac{f_i}{l_i^2} \quad (8)$$

$$b_i = -\frac{2f_i}{l_i} \quad (9)$$

$$y_i(x) = -\frac{f_i}{l_i^2}x^2 - \frac{2f_i}{l_i}x \quad (10)$$

$$y'_i(x) = -\frac{2f_i}{l_i^2}x - \frac{2f_i}{l_i} \quad (11)$$

Logo, no ponto em que $x = 0$, para a parábola em vermelho tem-se:

$$y'_i(0) = -\frac{2f_i}{l_i} = \tan \alpha \quad (12)$$

Para a curva em azul da Figura 4 determina-se:

$$y_s(x) = a_sx^2 + b_sx + c_s \quad (13)$$

$$y_s(0) = a_s \cdot 0 + b_s \cdot 0 + c_s = 0 \therefore c_s = 0 \quad (14)$$

$$y_s(l_s) = -f_s = a_sl_s^2 + b_sl_s \quad (15)$$

$$y'_s(x) = 2a_sx + b_s \quad (16)$$

$$y'_s(l_s) = 0 = 2a_sl_s + b_s \therefore b_s = -2a_sl_s \quad (17)$$

$$-f_s = a_sl_s^2 - 2a_sl_s^2 \therefore a_s = \frac{f_s}{l_s^2} \quad (18)$$

$$y_s(x) = \frac{f_s}{l_s^2}x^2 - \frac{2f_s}{l_s}x \quad (19)$$

$$y'_s(x) = \frac{2f_s}{l_s^2}x - \frac{2f_s}{l_s} \quad (20)$$

$$y_s(x) = a_s x^2 + b_s x + c_s \quad (21)$$

Logo, no ponto em que $x = 0$, para a parábola em azul tem-se:

$$y'_s(0) = -\frac{2f_s}{l_s} = \tan \alpha \quad (22)$$

Visto que a tangente do ângulo de um triângulo-retângulo é igual ao cateto oposto sobre o adjacente (altura / base), resulta que para ambas curvas encontramos $2f/l$ sendo $2f$ a altura do triângulo e l sua base. Uma vez comprovado isso, basta igualar as tangentes do ponto ou realizar uma proporção de triângulos que o resultado será o mesmo.

$$\tan \alpha = -\frac{2f_s}{l_s} = -\frac{2f_i}{l_i} \therefore f_i = \frac{l_i}{l_s} f_s \quad (23)$$

Lembrando que $\Delta h = f_i + f_s$, tem-se:

$$\Delta h = f_i + f_s \therefore f_s = \Delta h - f_i \quad (24)$$

$$f_i = \frac{l_i}{l_s} (\Delta h - f_i) \quad (25)$$

$$f_i + \frac{l_i}{l_s} f_i = \frac{l_i}{l_s} \Delta h \quad (26)$$

Logo, f_i será igual a:

$$f_i = \frac{l_i}{l_s} \frac{\Delta h}{\left(1 + \frac{l_i}{l_s}\right)} \quad (27)$$

Substituindo novamente f_i na primeira equação de Δh :

$$f_s = \Delta h - \frac{l_i}{l_s} \frac{\Delta h}{\left(1 + \frac{l_i}{l_s}\right)} = \frac{\Delta h \left(1 + \frac{l_i}{l_s}\right)}{\left(1 + \frac{l_i}{l_s}\right)} - \frac{\left(\frac{l_i}{l_s}\right) \Delta h}{\left(1 + \frac{l_i}{l_s}\right)} \quad (28)$$

Logo, f_s será igual a:

$$f_s = \frac{\Delta h}{\left(1 + \frac{l_i}{l_s}\right)} \quad (29)$$

4.2 GEOMETRIA DA SEÇÃO DA VIGA

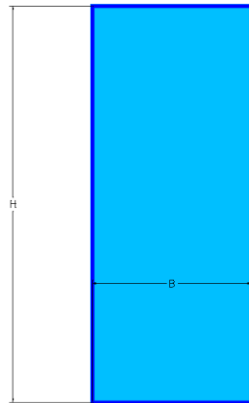
A geometria da seção determina diversos parâmetros necessários à resolução dos esforços da estrutura, bem como para a determinação das armaduras. O dimensionamento apropriado da estrutura depende de fatores como momento de inércia, área, centro de gravidade, forma, entre outros.

4.2.1 Do Cálculo das Propriedades

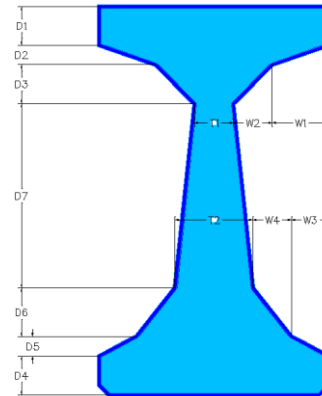
Para formatos padrão disponíveis na escolha da seção, o cálculo de propriedades ocorre de maneira exata, conforme cada geometria a partir de suas coordenadas geradas pelo programa. Com os algoritmos escolhidos para cálculo das propriedades da seção transversal, qualquer forma de seção seria possível, dependendo apenas de implementação adicional em termos de interface, ficando para uma implementação futura, de modo a permitir a inserção de formas quaisquer genéricas, desenhadas pelo próprio projetista.

Essas expressões baseiam-se nas informações geométricas das peças, sendo para cada tipo de seção os parâmetros correspondentes aos da Figura 5 que segue:

Figura 5 – Parâmetros geométricos da seção da viga por tipo de seção.



Seção tipo retangular: parâmetros b e h .



Seção tipo I: parâmetros $D1$, $D2$, $D3$, $D4$, $D5$, $D6$, $D7$, $W1$, $W2$, $W3$, $W4$, $T1$, $T2$ e $C1$.

(fonte: elaborada pelo autor.)

O detalhamento desses algoritmos e das propriedades calculadas serão apresentados em maior detalhe no capítulo 9.

5 PROTENSÃO DO CONCRETO

A protensão do concreto tem por objetivo melhorar seu desempenho, em especial, para vãos maiores do que aqueles possíveis com um concreto apenas armado. Isso é obtido através de uma armadura de aço para protensão, que produz um carregamento ativo contrário aos outros solicitantes na estrutura. Ele é ativo pois o aço que é tracionado e preso ao concreto tende a buscar sua forma inicial indeformada, gerando compressão e momentos ao longo do concreto que o impede de se deslocar. Como resultado da combinação da protensão na peça, a solicitação efetiva fica diminuída das forças contrárias geradas pela armadura ativa, permitindo assim que se obtenha melhor desempenho final em que as tensões no concreto reduzem, e por consequência sua flecha e fissuras.

Para determinação da força necessária a essa armadura, deve-se levar em conta fatores diversos, pois apesar de ser estática em um curto período de tempo (uns poucos dias), ela sofre variações no tempo ao longo da vida útil da estrutura. Essa perda se deve a fenômenos dos materiais empregados, sendo considerados a fluência e retração para o concreto, e a relaxação no aço.

5.1 TIPOS DE PROTENSÃO

A NBR 6118:2014 define quatro tipos de protensão, em função do desempenho e das cargas compensadas pela ação desta. A protensão pode ser completa, limitada, parcial, ou ainda de um quarto tipo adicional chamado de superprotensão, não tratado pela norma. Para determinação do tipo, a análise é feita com base nos tipos de carregamentos de serviço, frequentes, quase permanentes ou raros, e das exigências dos ELS determinados na Tabela 2 a seguir. Na protensão completa respeita-se o ELS de descompressão (ELS-D) para carregamentos quase permanentes, ou seja, tração nula; na limitada, o ELS de formação de fissuras (ELS-W) admite-se uma certa tração no concreto para combinações frequentes, limitada em até $1,20 \cdot f_{cm}$; na protensão parcial respeita-se o ELS para abertura de fissuras, não sendo definido um limite específico para tração no concreto, desde que w_k fique no máximo em 0,2 mm; a superprotensão

ocorre quando mesmo na parte menos comprimida do concreto ainda existir compressão, sendo especificado de acordo com cada tipo de obra ou projeto.

Tabela 2 – Exigências de durabilidade relacionadas à fissuração e à proteção da armadura, em função das classes de agressividade ambiental.

Tipo de concreto estrutural	Classe de agressividade ambiental (CAA) e tipo de protensão	Exigências relativas à fissuração	Combinação de ações em serviço a utilizar
Concreto protendido nível 1 (protensão parcial)	Pré-tração com CAA I ou Pós-tração com CAA I e II	ELS-W $w_k \leq 0,2$ mm	Combinação frequente
Concreto protendido nível 2 (protensão limitada)	Pré-tração com CAA II ou Pós-tração com CAA III e IV	Verificar as duas condições abaixo	
		ELS-F	Combinação frequente
		ELS-D ^a	Combinação quase permanente
Concreto protendido nível 3 (protensão completa)	Pré-tração com CAA III e IV	Verificar as duas condições abaixo	
		ELS-F	Combinação rara
		ELS-D ^a	Combinação frequente
^a A critério do projetista, o ELS-D pode ser substituído pelo ELS-DP com $a_p = 50$ mm.			

(fonte: adaptado da NBR 6118/2014.)

A protensão pode ser realizada por pré-tração ou pós-tração de cabos ou cordoalhas inseridas no concreto. Quando a tração da armadura se dá no concreto ainda fresco chama-se de pré-tração, que será fixada pelo próprio concreto quando endurecido, de modo que a força de tração do aço se transfere ao concreto por aderência entre esses materiais. Quando a tração da armadura se dá no concreto já endurecido chama-se de pós-tração. No caso da pós-tração, as armaduras podem estar em contato direto com o concreto endurecido, sendo tracionadas e mantidas na posição final pelo atrito resultante com o concreto, denominado de pós-tração aderente, ou estar inseridas dentro de mangueiras no concreto, engraxadas ou não para que não haja atrito da armadura com o concreto em si, de modo que a transferência da força se dá pela ancoragem do aço nas extremidades da peça, denominando-se assim de pós-tração não aderente esse método.

5.2 MATERIAIS

O concreto e o aço devem trabalhar em compatibilidade de deformações para os casos em que existe aderência da armadura (pré-tração ou pós-tração aderente), de modo que os limites entre esses materiais devem ser respeitados no conjunto.

5.2.1 Tensão do Concreto

A tensão de compressão do concreto nos Estados Limites de Serviço (ELS) encontra-se geralmente em uma região ainda quase-linear, pois relativamente baixas em relação ao Estado Limite Último (ELU). Desse modo, a lei de Hooke é empregada para tensões de compressão menores que $0,5f_c$, com uma pequena aproximação para o módulo de elasticidade do concreto por seu secante, em NBR 6118 (2014) conforme segue:

$$\sigma_c = E_{cs} \cdot \varepsilon_c \quad (30)$$

Onde,

σ_c : tensão de compressão do concreto (kN/m²);

E_{cs} : módulo de elasticidade secante do concreto (kN/m²);

ε_c : deformação do concreto (-).

O módulo de elasticidade secante do concreto pode ser calculado pela expressão em NBR 6118 (2014) a seguir:

$$E_{cs} = \alpha_i \cdot E_{ci} \quad (31)$$

Onde:,

E_{cs} : módulo de elasticidade secante do concreto;

α_i : fator de interpolação linear;

E_{ci} : módulo de elasticidade inicial do concreto.

O fator de interpolação linear inicial é dado por norma NBR 6118 (2014) como:

$$\alpha_i = 0,8 + 0,2 \cdot \frac{f_{ck}}{80} \leq 1,0 \quad (32)$$

Onde,

α_i : fator de interpolação linear;

f_{ck} : resistência característica do concreto à compressão (em MPa).

O módulo de elasticidade inicial do concreto é definido por duas expressões distintas, dependente da resistência característica do concreto à compressão, em NBR 6118 (2014):

$$E_{ci} = \alpha_E \cdot 5000 \sqrt{f_{ck}}, \text{ para } f_{ck} \text{ de } 20 \text{ MPa a } 50 \text{ MPa} \quad (33)$$

$$E_{ci} = 21,5 \cdot 10^3 \cdot \alpha_E \cdot \left(\frac{f_{ck}}{10} + 1,25 \right)^{\frac{1}{3}}, \text{ para } f_{ck} \text{ de } 55 \text{ MPa a } 90 \text{ MPa} \quad (34)$$

Onde:

E_{ci} : módulo de elasticidade inicial do concreto;

α_E : fator de dependência do agregado no concreto.

Os valores para o fator α_E estão apresentados a seguir, da NBR 6118 (2014):

- a) $\alpha_E = 1,2$ (para basalto e diabásio)
- b) $\alpha_E = 1,0$ (para granito e gnaisse)
- c) $\alpha_E = 0,9$ (para calcário)
- d) $\alpha_E = 0,7$ (para arenito)

No Estado Limite Último (ELU), a tensão de compressão divide-se em função da deformação no concreto, em faixas de 0 ‰ até 2 ‰ e de 2 até 3,5 ‰, para concretos até f_{ck} de 50 MPa, por exemplo, quando nesta última considera-se então a ruptura do material. Para concretos acima dos 50 MPa, esses valores são obtidos pelas equações (39) e (40). No primeiro intervalo o comportamento da tensão por deformação é aproximado por uma parábola, ao passo que por

simplificação o comportamento para o segundo intervalo é uma constante. Assim, as equações desse comportamento são descritas na NBR 6118 (2014) como:

$$\sigma_c = 0,85 \cdot f_{cd} \cdot \left[1 - \left(1 - \frac{\varepsilon_c}{\varepsilon_{c2}} \right)^n \right], \text{ para } 0 \leq \varepsilon_c \leq \varepsilon_{c2} \quad (35)$$

$$\sigma_c = f_c, \text{ para } \varepsilon_{c2} \leq \varepsilon_c \leq \varepsilon_{cu} \quad (36)$$

Onde,

σ_c : tensão no concreto;

f_c : limite de compressão do concreto calculado como $0,85f_{ck}/\gamma_c$;

ε_c : deformação do concreto;

ε_{c2} : deformação específica de encurtamento do concreto no início do patamar plástico;

ε_{cu} : deformação no limite de ruptura do concreto;

n : fator de ajuste em função de f_{ck} , apresentado nas expressões (37) e (38).

O fator n pode ser calculado pelas expressões abaixo, da NBR 6118 (2014):

$$n = 2, \text{ para } f_{ck} \leq 50 \text{ MPa} \quad (37)$$

$$n = 1,4 + 23,4[(90 - f_{ck})/100]^4, \text{ para } f_{ck} > 50 \text{ MPa} \quad (38)$$

O ε_{c2} e ε_{cu} , adaptado pelo autor da NBR 6118 (2014), são dados por:

$$\varepsilon_{c2} = 2,0 \text{ ‰} + 0,085 \text{ ‰} \cdot (f_{ck} - 50)^{0,53}, \text{ sendo } f_{ck} - 50 \geq 0 \quad (39)$$

$$\varepsilon_{c2} = 2,6 \text{ ‰} + 35 \text{ ‰} \cdot [(90 - f_{ck})/100]^4, \text{ sendo } 90 - f_{ck} \leq 40 \quad (40)$$

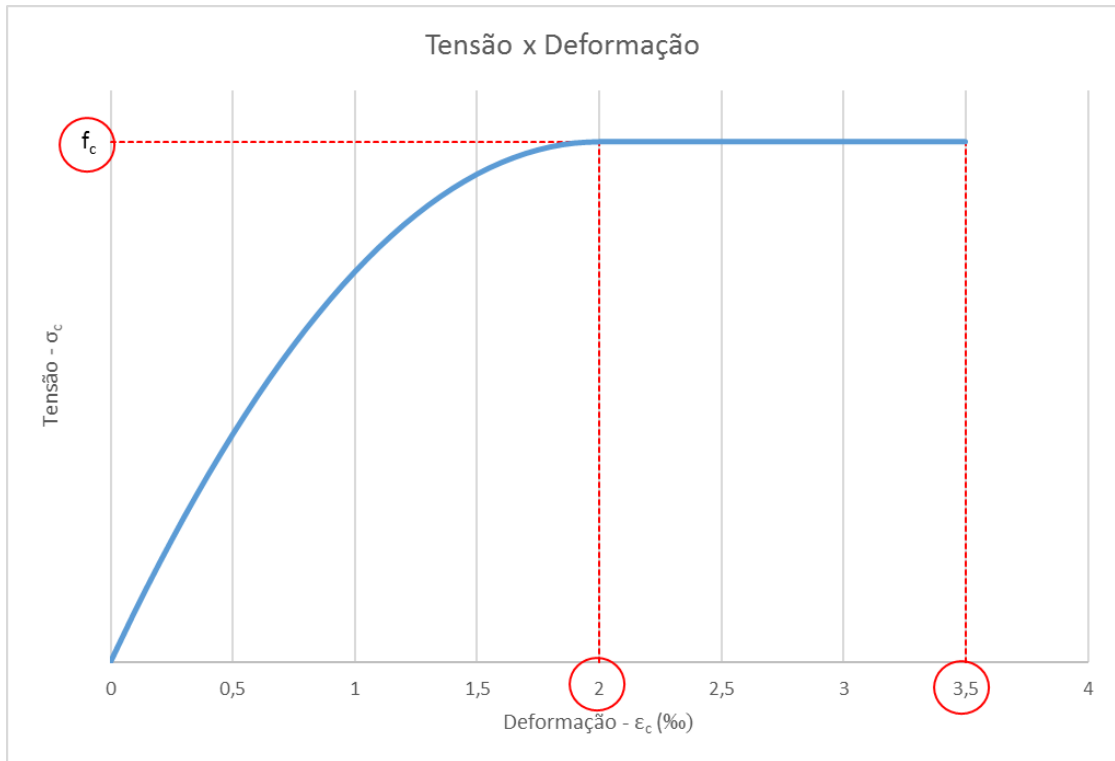
Onde,

ε_{c2} : deformação específica de encurtamento do concreto no início do patamar plástico;

f_{ck} : resistência característica do concreto à compressão (em MPa).

O resultado gráfico da equação (35) está ilustrado na Figura 6, para concretos até $f_{ck} = 50$ MPa.

Figura 6 – Diagrama tensão-deformação idealizado.



(fonte: elaborada pelo autor.)

Para a tração do concreto consideram-se as expressões de $f_{ctk,inf}$, $f_{ctk,sup}$ e $f_{ct,m}$ conforme norma. As fórmulas para elas são dadas pela NBR 6118 (2014) como:

$$f_{ctk,inf} = 0,7 f_{ct,m} \quad (41)$$

$$f_{ctk,sup} = 1,3 f_{ct,m} \quad (42)$$

$$f_{ct,m} = 0,3 f_{ck}^{2/3}, \text{ para concretos de classes até C50} \quad (43)$$

$$f_{ct,m} = 2,12 \ln(1 + 0,11 f_{ck}), \text{ para concretos de classes C55 até C90} \quad (44)$$

Onde,

$f_{ctk,inf}$: limite inferior da resistência característica de tração do concreto;

$f_{ctk,sup}$: limite superior da resistência característica de tração do concreto;

$f_{ct,m}$: resistência média de tração do concreto;

f_{ck} : resistência característica do concreto à compressão (em MPa).

5.2.2 Aço Para Armadura Passiva

O aço disponível inicialmente no programa para armaduras passivas será somente o CA-50, pois este é o mais comumente utilizado em vigas, podendo-se adicionar outros aços futuramente, caso surja a necessidade para tanto. Não obstante, o projetista poderá alterar o valor característico resistente desse aço para melhor adequar às suas necessidades específicas para cada projeto.

O módulo de elasticidade utilizado é de 210 GPa, conforme recomendação de norma brasileira, podendo ser alterado no programa para cada projeto.

O diagrama de tensão-deformação utilizado para cálculo da resistência ao escoamento e à tração desse aço utilizado é simplificado conforme Figura 7. O limite de deformação considerado para escoamento do aço é de 2 ‰, sendo o limite máximo de 10 ‰ considerado em função do concreto, pois estimam-se fissuras de 0,3 a 0,4 mm para estes valores.

A expressão para cálculo utilizada da resistência de projeto do aço para armaduras passivas na NBR 6118 (2014) é:

$$f_{yd} = f_{yk}/\gamma_s \quad (45)$$

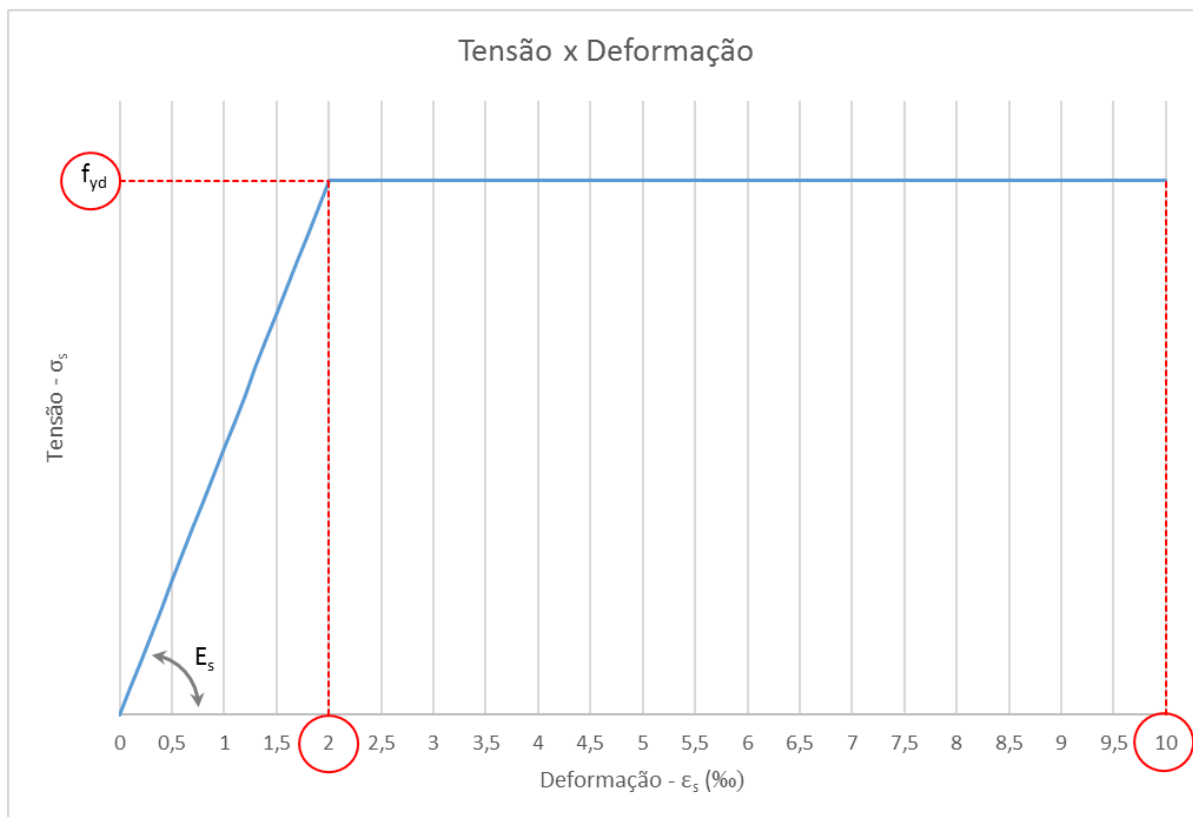
Onde,

f_{yd} : resistência de projeto do aço para armaduras passivas;

f_{yk} : resistência característica do aço para armaduras passivas;

γ_s : coeficiente de segurança do aço para armaduras passivas, igual a 1,15.

Figura 7 – Diagrama tensão deformação para aços de armaduras passivas.



(fonte: elaborada pelo autor.)

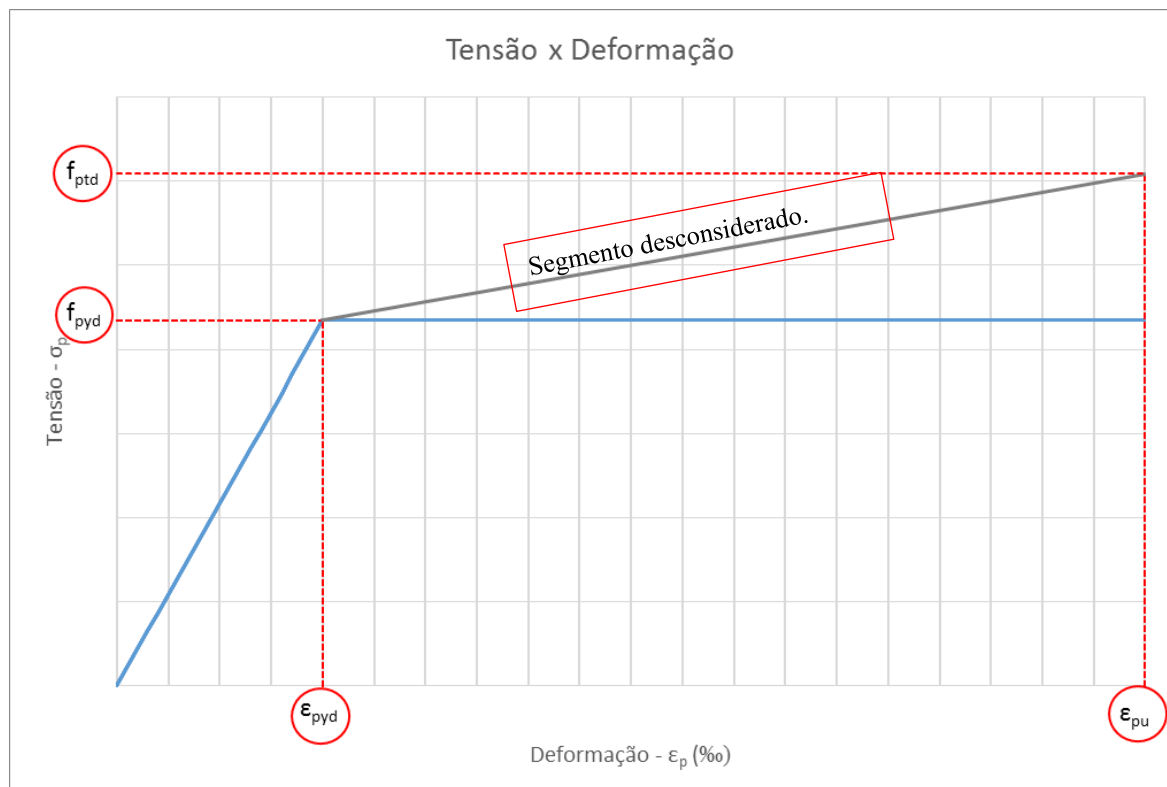
5.2.3 Aço da Armadura Ativa

O módulo de elasticidade das armaduras de protensão seguem um padrão de 200 GPa, conforme recomendação da NBR 6118 (2014), no caso da falta de dados específicos do fabricante. No entanto, esse valor pode ser alterado pelo projetista, caso esse disponha de melhores informações para o uso no dimensionamento das armaduras ativas.

O cálculo da tensão no aço da armadura ativa é feito de maneira simplificada em relação ao diagrama tensão-deformação apresentado na NBR 6118 (2014). Na norma, o aço da armadura de protensão possui dois comportamentos lineares, antes e após o escoamento, em que sua tangente de inclinação reduz, possuindo certa declividade mesmo após o escoamento. Para esse trabalho adotou-se uma simplificação de desconsiderar esse ganho de tensão após o escoamento da protensão, a favor da segurança.

A Figura 8 apresenta uma comparação do diagrama exposto na NBR 6118 (2014) e do considerado na ferramenta desenvolvida para o cálculo de vigas protendidas desse trabalho.

Figura 8 – Diagrama tensão deformação para aços de armaduras ativas.



(fonte: elaborada pelo autor.)

O valor de cálculo da resistência do aço para armadura ativa é dependente não apenas do seu valor característico e de um coeficiente de segurança, mas também da forma de aplicação da protensão, isto é, pré-tração ou pós-tração, e da classe do aço (relaxação baixa – RB, ou relaxação normal – RN). As expressões para determinar a resistência de cálculo para cada um desses aços, segundo NBR 6118 (2014) são:

$$f_{pyd} = \frac{0,85 \cdot f_{ptk}}{\gamma_s}, \text{ para classe RN} \quad (46)$$

$$f_{pyd} = \frac{0,9 \cdot f_{ptk}}{\gamma_s}, \text{ para classe RB} \quad (47)$$

Onde,

f_{pyd} : resistência de cálculo do aço de protensão;

f_{ptk} : resistência característica de tração do aço de protensão;

γ_s : coeficiente de segurança para dimensionamento de armaduras, de valor 1,15.

O limite de deformação para escoamento do aço é calculado pela NBR 6118 (2014) por:

$$\varepsilon_{pyd} = \frac{f_{pyd}}{E_p} \quad (48)$$

Onde,

ε_{pyd} : limite de deformação para escoamento do aço;

f_{ptk} : resistência característica de tração do aço de protensão;

E_p : módulo de deformação elástica do aço de protensão.

6 DETERMINAÇÃO DA PROTENSÃO

O dimensionamento da protensão deve levar em conta um máximo aproveitamento do aço da armadura ativa, respeitando os limites de tensões que o concreto pode suportar sem romper. Ainda deve-se considerar qual a parcela de carga que se deseja absorver por esse tipo de armadura. Na questão de desempenho da peça em termos de flechas e fissuras, deve-se observar que a protensão aplicada inicialmente sofrerá diferentes perdas, tanto imediatas quanto ao longo do tempo, devendo-se levar em conta também esses valores para avaliar se os resultados são satisfatórios ou se exige redimensionar a armadura.

6.1 MÉTODO DO BALANCEAMENTO DE CARGAS

Segundo T. Y. Lin, os esforços gerados pela armadura de protensão dentro da viga, para cabos parabólicos, pode ser substituída por cargas distribuídas equivalentes para cada trecho de parábola. Desta forma, é possível correlacionar os esforços solicitantes com os esforços gerados pela protensão, buscando-se um equilíbrio de cargas com o objetivo de reduzir efeitos indesejados e melhorar o desempenho da estrutura, conforme mencionado no capítulo 6 anteriormente.

A obtenção dessas cargas equivalentes pode ser demonstrada pelas seguintes relações fundamentais da mecânica estrutural:

$$Q(x) = - \int f(x). dx \quad (49)$$

$$M(x) = - \iint f(x). dx^2 \quad (50)$$

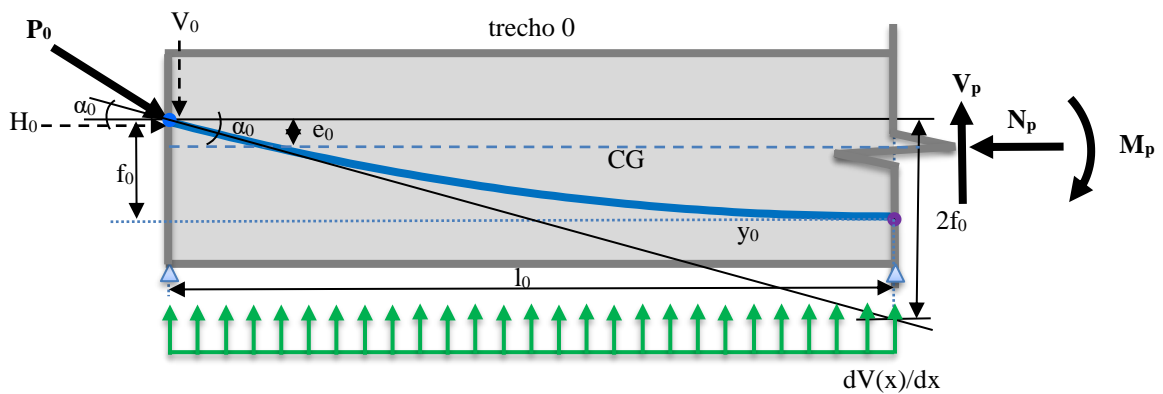
Para o caso de uma carga distribuída constante, a que se dedica esse método, tem-se que $f(x)=q$, e assim as relações resultam em:

$$Q(x) = -q.x + cte_1 \quad (51)$$

$$M(x) = -\frac{q}{2} \cdot x^2 + cte_1 \cdot x + cte_2 \quad (52)$$

Note-se que a função de momento resultante é uma parábola. Agora para a protensão, temos que os $y's(x)$ determinados na geometria do cabo de protensão, conforme explicitado na seção Determinação das equações das parábolas, são parábolas que descrevem a excentricidade do cabo, pois estão referenciadas ao centro de gravidade da seção. Dessa forma, pode-se determinar o momento causado pela protensão a uma seção qualquer pelo seguinte:

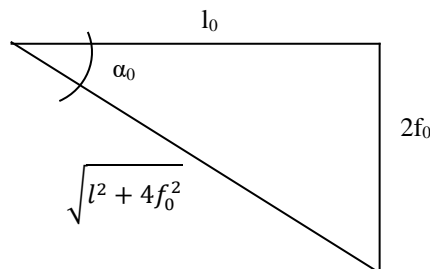
Figura 9 – Solicitações internas na viga devido à protensão.



(fonte: elaborada pelo autor.)

Logo, geometricamente tem-se que:

Figura 10 – Relações trigonométricas com ângulo inicial e forma da parábola.



(fonte: elaborada pelo autor.)

$$\operatorname{tg} \alpha_0 = \frac{2f_0}{l_0} \quad (53)$$

$$\operatorname{sen} \alpha_0 = \frac{2f_0}{\sqrt{l_0^2 + 4f_0^2}} \quad (54)$$

$$\operatorname{cos} \alpha_0 = \frac{l_0}{\sqrt{l_0^2 + 4f_0^2}} \quad (55)$$

Logo,

$$V_0 = P_0 \cdot \operatorname{sen} \alpha_0 = P_0 \cdot \frac{2f_0}{\sqrt{l_0^2 + 4f_0^2}} \approx P_0 \cdot \operatorname{tg} \alpha_0 = P_0 \cdot \alpha_0 \quad (56)$$

$$H_0 = P_0 \cdot \operatorname{cos} \alpha_0 = P_0 \cdot \frac{l_0}{\sqrt{l_0^2 + 4f_0^2}} \approx P_0 \cdot 1 = P_0 \quad (57)$$

Essa simplificação significa que os resultados obtidos somente serão válidos se α_0 for muito pequeno, adotando-se assim $\operatorname{sen} \alpha_0 = \operatorname{tg} \alpha_0 = \alpha_0$, $\operatorname{cos} \alpha_0 = 1$.

Realizando-se o equilíbrio de forças para a Figura 9, pode-se obter a seguinte expressão para as solicitações geradas pela protensão, sendo que:

$$\operatorname{tg} \alpha(x) = \frac{dy_0(x)}{dx} \approx \alpha(x) = -\frac{2f_0}{l_0^2} x + \frac{2f_0}{l_0} \quad (58)$$

$$V(x) = P_0 \cdot \alpha(x) = P_0 \left(-\frac{2f_0}{l_0^2} x + \frac{2f_0}{l_0} \right) \quad (59)$$

$$\frac{dV(x)}{dx} = -\frac{2f_0}{l_0^2} P_0 \quad (60)$$

Assim, somando-se as forças em verticais ao longo de x , obtém-se que:

$$V_p(x) = V_0 + \int \frac{dV(x)}{dx} dx \quad (61)$$

$$V_p(x) = P_0 \cdot \frac{2f_0}{l_0} + \left(-\frac{2f_0}{l_0^2} P_0 \right) x \quad (62)$$

$$V_p(x) = P_0 \frac{2f_0}{l_0^2} (l_0 - x) \quad (63)$$

Com a soma das forças horizontais ao longo de x, obtém-se que:

$$N_p(x) = P_0 \cdot \cos(x) = P_0 \quad (64)$$

Pela equação de equilíbrio de momentos, pode-se obter também:

$$M_p(x) = -V_0 \cdot x + \int \frac{dV(x)}{dx} x \cdot dx \quad (65)$$

$$M_p(x) = -P_0 \cdot \frac{2f_0}{l_0} \cdot x + \int -\frac{2f_0}{l_0^2} P_0 \cdot x \cdot dx = P_0 \cdot \frac{f_0}{l_0^2} (lx - x^2) = P_0 \cdot y_0(x) \quad (66)$$

Visto que $y_0(x)$ é a excentricidade do cabo de protensão, $M_p(x)$ é simplesmente a força P de protensão do cabo pela excentricidade do mesmo.

Para exemplificar a validade das simplificações, pode-se comparar o resultado exato das forças verticais e horizontais de entrada da protensão na viga, com aquelas simplificadas para diferentes flechas e comprimentos de viga:

Tabela 3 – Comparação das projeções de valores da protensão por cálculo exato e por simplificação.

Valores Exatos

f [L]	l [L]	P ₀ [F]	V ₀ [F]	H ₀ [F]	f/l [-]	α ₀ (graus)	α ₀ (rad)	tg α ₀	sen α ₀	cos α ₀
0,3	6	1000	196,1161	980,5807	5,0%	11,31	0,1974	0,2000	0,1961	0,9806
0,3	10	1000	119,1452	992,8768	3,0%	6,84	0,1194	0,1200	0,1191	0,9929
0,3	15	1000	79,74522	996,8153	2,0%	4,57	0,0798	0,0800	0,0797	0,9968

Valores por Aproximação

f [L]	l [L]	P ₀ [F]	V ₀ [F]	H ₀ [F]	f/l [-]	α ₀ (graus)	α ₀ (rad)	tg α ₀	sen α ₀	cos α ₀
0,3	6	1000	197,3956	1000	5,0%	11,31	0,1974	0,1974	0,1974	1,0000
0,3	10	1000	119,4289	1000	3,0%	6,84	0,1194	0,1194	0,1194	1,0000
0,3	15	1000	79,82999	1000	2,0%	4,57	0,0798	0,0798	0,0798	1,0000

(fonte: elaborada pelo autor.)

Pela Tabela 3 é possível observar a importância dessa premissa de que o ângulo da protensão será muito pequeno para que seja válido o método aplicado. A prática da indústria mostra que essa metodologia é válida na maioria dos casos pois de fato as geometrias das protensões projetadas são adequadas à aplicação dessa simplificação.

Com base na equação (66) obtida, em que se relaciona o momento de protensão com a força de tração introduzida no cabo pela equação da parábola que descreve a geometria do cabo, pode-se aplicar as relações fundamentais apresentadas para determinar-se a carga equivalente produzida na viga, conforme segue:

$$Q(x) = -q \cdot x + cte_1 \quad (67)$$

$$\frac{Q(x)}{dx} = -q \quad (68)$$

$$M(x) = - \iint f(x) \cdot dx^2 = - \iint q \cdot dx^2 \quad (69)$$

$$\frac{d(M(x))}{dx^2} = \frac{d(- \iint q \cdot dx^2)}{dx^2} = -q \quad (70)$$

$$\frac{d(M_P)}{dx^2} = \frac{d\left(P_0 \cdot \frac{f_0}{l_0^2} (lx - x^2)\right)}{dx^2} = -\frac{2f_0}{l_0^2} P_0 = -q_{Peq} \quad (71)$$

$$q_{Peq} = \frac{2f_0}{l_0^2} P_0 \quad (72)$$

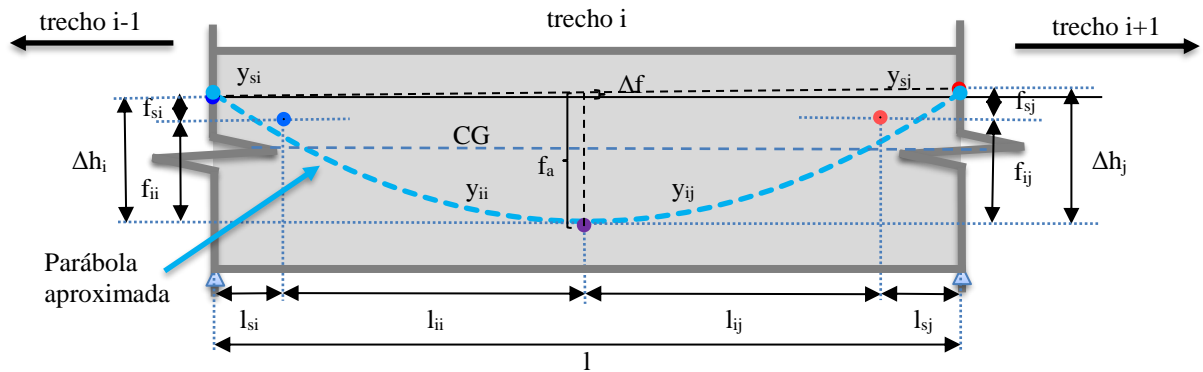
6.2 PROTENSÃO EQUIVALENTE EM CADA TRECHO

Conforme demonstrado em na seção anterior, para cada trecho de parábola pode-se determinar uma carga equivalente produzida pela protensão, podendo-se substituir a protensão da viga por diferentes cargas distribuídas, simplificando sua análise.

Para equalizar as cargas produzidas pela protensão com aquelas das solicitações que se pretende equilibrar pela protensão, é possível ainda uma outra simplificação, em que para cada vão da

viga. Ao invés de considerar-se até quatro cargas por trecho para cada segmento de parábola, é possível criar uma parábola virtual com flecha aproximada, podendo então comparar-se uma única carga solicitante com o equilíbrio que será introduzido pela geometria da protensão.

Figura 11 – Definição de uma parábola equivalente para o trecho para simplificação das cargas.



(fonte: elaborada pelo autor.)

Pela relação de triângulos, f_a será o menor dos Δh mais um Δf , conforme segue:

$$f_a = \Delta f + \text{menor}(\Delta h) \quad (73)$$

$$\Delta f = \frac{\text{maior}(\Delta h) - \text{menor}(\Delta h)}{l} \cdot \frac{l}{2} = \frac{\text{maior}(\Delta h) - \text{menor}(\Delta h)}{2} \quad (74)$$

$$f_a = \Delta f + \text{menor}(\Delta h) \quad (75)$$

$$f_a = \frac{\text{maior}(\Delta h) - \text{menor}(\Delta h)}{2} + \text{menor}(\Delta h) \quad (76)$$

Por fim, a carga equivalente para o trecho i, produzida pela protensão, será igual a:

$$q_{Peq_{trecho}} = \frac{8f_a}{l^2} P_{trecho} = q_{solicitante_{trecho}} \quad (77)$$

Dessa forma pode-se obter uma força de protensão P estimada para cada trecho:

$$P_{trecho} = \frac{l^2}{8f_a} q_{solicitante_{trecho}} \quad (78)$$

6.3 PROTENSÃO DA VIGA COMO UM TODO

Para este trabalho foi considerado que a armadura de protensão será única ao longo de toda a viga, de modo que somente pode ser aplicada uma única força de protensão para todos os trechos. Dessa forma, conforme explicado na seção sobre a força de protensão equivalente para cada vão da viga, essa força deverá ser única ao final do dimensionamento. Dessa forma, a metodologia criada para compatibilização da protensão entre os trechos é de que dentre todos os valores calculados para os trechos, todos com flecha máxima respeitando as coberturas de concreto informadas para esse tipo de armadura, o maior deverá ser tomado como resultado inicial da força dimensionada. Após, os demais trechos que tiveram uma força P_{trecho} inferior calculada, deverão ter sua flecha adaptada (reduzida) para que a carga equivalente se mantenha igual àquela necessária para o trecho. Assim, uma única força de protensão é obtida ao final desse processo, com as flechas ajustadas para cada trecho em uma geometria mais próxima da final, ficando pendente as verificações em Estado Limite de Serviço e Estado Limite Último, que poderá causar ainda alguns ajustes na protensão necessária no projeto final da viga.

Para o ajuste das flechas nos trechos, visto que a relação P_{trecho} por f_a é linear (eq. 78), e mantendo-se a mesma carga solicitante utilizada inicialmente, basta multiplicar a flecha de cada trecho por um fator $P_{trecho}/P_{máx}$, de modo a obter-se a geometria final do cabo.

7 VERIFICAÇÕES DA VIGA

Após definida uma tensão para protensão nos cabos, as tensões que estes produzem ao longo de toda a viga devem ser verificados com relação às tensões resistentes dos materiais para que estes não sofram esforços excessivos. Assim, as tensões admissíveis do material terão de ser definidas para comparação.

7.1 TENSÕES ADMISSÍVEIS DO CONCRETO

As tensões resistentes limite do concreto em suas fibras mais externas, inferiores e superiores, devem ser comparadas com as tensões geradas pelos carregamentos e protensão nos diferentes instantes de tempo ao longo da vida útil da estrutura, de um tempo inicial t_0 a um tempo t_∞ (equivalente a 10.000 dias).

O programa desenvolvido possibilita a definição de diferentes instantes de tempo que podem ser associados às cargas solicitantes sobre a viga. Dessa forma, cada carregamento será considerado em seu tempo por sua duração específica, de modo que possibilitará implementar no programa a criação de um número necessário de verificações em função do número de combinações temporais.

A resistência à compressão do concreto para suas diferentes idades é considerada pela NBR 6118 (2014) como:

- a) para verificação em data j inferior a 28 dias, adota-se:

$$f_{cd} = \frac{f_{ckj}}{\gamma_c} \quad (79)$$

$$f_{ckj} = \beta_1 f_{ck} \quad (80)$$

$$\beta_1 = e^s \left[1 - \left(\frac{28}{t} \right)^{\frac{1}{2}} \right] \quad (81)$$

b) para verificação em data j igual ou superior a 28 dias, adota-se f_{ck} .

$$f_{cd} = \frac{f_{ck}}{\gamma_c} \quad (82)$$

Onde,

f_{cd} é a resistência a compressão de cálculo do concreto, na idade desejada;

f_{ckj} é a resistência a compressão característica do concreto, na idade desejada;

f_{ck} é a resistência a compressão característica do concreto;

$s = 0,38$ para concreto de cimento CPM e IV;

$s = 0,25$ para concreto de cimento CPI e II;

$s = 0,20$ para concreto de cimento CPV-ARI;

t é a idade efetiva do concreto, expressa em dias

γ_c é o coeficiente de ponderação da resistência do concreto no ELU, conforme NBR 6118.

A resistência à tração do concreto para suas diferentes idades é considerada pela NBR 6118 (2014) como:

$$f_{ctd} = \frac{f_{ctk,inf}}{\gamma_c} \quad (83)$$

$$f_{ctk,inf} = 0,7f_{ct,m} \quad (84)$$

$$f_{ct,m} = 0,3f_{ckj}^{2/3} \quad (85)$$

com $f_{ckj} = f_{ck}$ para idades superiores a 28 dias

Onde,

f_{ctkj} – o mesmo calculado para a compressão, devendo ser maior ou igual a 7 MPa, e expresso em megapascal (MPa);

$f_{ct,m}$ – resistência média do concreto à tração;

$f_{ctk,inf}$ – resistência à tração do concreto, em seu limite inferior;

f_{ctd} – resistência de cálculo à tração do concreto;

γ_c – coeficiente de ponderação da resistência do concreto no ELU, conforme NBR 6118.

Os limites do concreto são definidos para tração e para compressão pela NBR 6118 (2014), em função do tempo, conforme expressões abaixo:

e) para idades do concreto até 28 dias:

$$\sigma_{ccj} \leq 0,70 f_{ckj} \quad (86)$$

$$\sigma_{ctj} \leq 1,20 f_{ct,m} \quad (87)$$

f) para demais idades, adota-se:

$$\sigma_{cc\infty} \leq 0,70 f_{cd} \quad (88)$$

$$\sigma_{ct\infty} \leq \begin{cases} 1,20 f_{ct,m} & (\text{protensão limitada}) \\ 0 & (\text{protensão completa}) \\ \text{outros limites} & (\text{protensão parcial}) \end{cases} \quad (89)$$

Onde,

σ_{ccj} – limite de tensão de compressão do concreto em uma idade j , até o limite de 28 dias;

σ_{ctj} – limite de tensão de tração do concreto em uma idade j , até o limite de 28 dias;

$\sigma_{cc\infty}$ – limite de tensão de compressão do concreto em um tempo considerado infinito, que representa uma idade do concreto a partir de 28 dias em diante;

$\sigma_{ct\infty}$ – limite de tensão de tração do concreto em um tempo considerado infinito, que representa uma idade do concreto a partir de 28 dias em diante.

Deve ser realizada a verificação das solicitações decorrentes dos carregamentos e protensão em cada instante de tempo especificado pelo projetista, e então comparar com os limites estabelecidos aqui para garantir a integridade e durabilidade da estrutura.

O cálculo das tensões é realizado para esforços de compressão e de flexão nesse método, considerando a protensão, peso próprio da estrutura, e demais carregamentos. A flexão irá produzir esforços máximos e diferentes nas fibras superiores e inferiores da viga, de modo que as verificações são realizadas para os extremos, tensões na borda superior e tensões na borda inferior da viga, versus a resistência dos materiais.

Em geral haverá sempre um instante de tempo inicial $t = t_0$ e um instante de tempo final $t = t_\infty$ a considerar. O instante t_0 poderá definido pelo projetista, correspondente a uma resistência mínima para possibilitar a aplicação da protensão na viga, ou por padrão será considerado igual a 15 dias. O tempo t_∞ é para uma idade de 10.000 dias, sem possibilidade de modificação, nem da criação de outros instantes de tempo superiores a esse pelo usuário.

As tensões solicitantes deverão ficar entre os valores limites para cada instante de tempo verificado, considerando as perdas da protensão até aquele instante, o peso próprio da viga e os demais carregamentos a cada posição desta até aquele instante, conforme as expressões genéricas descritas abaixo para borda superior (B.S.) e borda inferior (B.I.) da viga:

$$B.S.: \quad \sigma_{cc}(t) \leq P(t, x) \left(-\frac{1}{A_c} + \frac{e_p(x)}{W_s} \right) - \frac{M_g(x) + M_q(t, x)}{W_s} \leq \sigma_{ct}(t) \quad (90)$$

$$B.I.: \quad \sigma_{cc}(t) \leq P(t, x) \left(-\frac{1}{A_c} - \frac{e_p(x)}{W_i} \right) + \frac{M_g(x) + M_q(t, x)}{W_i} \leq \sigma_{ct}(t) \quad (91)$$

Onde,

$\sigma_{cc}(t)$ – limite de tensão de compressão do concreto protendido para um dado instante de tempo t (kN/m²);

$\sigma_{ct}(t)$ – limite de tensão de tração do concreto protendido para um dado instante de tempo t (kN/m²);

$P(t,x)$ – força de protensão efetiva da armadura (após perdas) em uma dada posição de abscissa x ao longo da viga, para um dado instante de tempo t (kN);

$1/A_c$ – fator de multiplicação de $P(t,x)$ para obtenção da tensão de compressão relativa ao esforço normal produzido pela protensão da viga em uma dada seção x para um instante de tempo t ;

A_c – área da seção transversal de concreto (m²);

$e_p(x)/W_{(s \text{ ou } i)}$ – fator de multiplicação de $P(t,x)$ para obtenção tensão de tração/compressão relativo ao esforço de flexão produzido pela protensão em uma dada seção x da viga em um instante de tempo t ;

$e_p(x)$ – excentricidade da armadura ativa em relação ao CG da viga na posição x (m);

W_s – módulo de resistência à flexão acima do CG da viga (m³);

W_i – módulo de resistência à flexão abaixo do CG da viga (m³);

$M_g(x)$ – momento produzido pelo peso próprio da estrutura em uma abscissa x ao longo da viga (kN.m);

$M_q(t, x)$ – momento produzido pelas cargas variáveis em um dado instante de tempo t e em uma abscissa x ao longo da viga (kN.m).

8 PERDAS DA FORÇA DE PROTENSÃO

A força de protensão aplicada inicialmente na armadura ativa da viga sofrerá processos de perdas tanto imediatas como dependentes da passagem do tempo. De maneira genérica, a força de protensão a cada instante e a qualquer posição da viga pode ser descrita por, segundo VERÍSSIMO (1998):

$$P(t, x) = P_0(x) - \Delta P(t, x) = P_i - \Delta P_0(x) - \Delta P(t, x) \quad (92)$$

Essa expressão é a mesma constante na NBR 6118:2014, porém com a notação adaptada para explicitar a dependência das funções em relação à posição ao longo da viga, x , e em relação ao tempo, t .

A variável $\Delta P_0(x)$ representa aquelas perdas imediatas, independentes do tempo porém dependentes da posição da viga, e a $\Delta P(t, x)$ representa as perdas dependentes do tempo para ocorram, e que também dependem da posição da viga em que esse está calculando. A descrição das componentes desses dois tipos de perdas, bem como seu significado e equacionamento serão apresentados ao longo desse capítulo.

8.1 PERDAS IMEDIATAS

As perdas da força de protensão são decorrentes de diversos fatores, e dependem do tipo da protensão realizada. No caso deste trabalho de pós-tração, as perdas imediatas a serem consideradas são decorrentes do encurtamento elástico do concreto, atrito e recuo de ancoragens. Cada um desses elementos será melhor descrito a seguir.

8.1.1 Perda Imediata por Encurtamento Elástico do Concreto

Neste trabalho tratamos do uso de pós-tração do aço de protensão, com cordoalhas engraxadas, isto é, não existe aderência entre o concreto e o aço, de modo que se deve realizar a

compatibilização dos deslocamentos do aço de protensão e do concreto, e não das tensões ao longo deles. Desse modo a premissa inicial é de que:

$$\Delta L_{cp} = \Delta L_p \quad (93)$$

Onde,

ΔL_p – variação do comprimento da armadura de protensão;

ΔL_{cp} – variação do comprimento da peça de concreto devido à protensão.

O esforço normal provocado pela protensão sobre a viga causa um encurtamento do concreto, conforme lei de Hooke, devido à presença da tensão de compressão produzida. Esse encurtamento será progressivo, à medida que cada um dos cabos for sendo tracionado, de modo que para o primeiro cabo, inicialmente, a tensão aplicada já será a efetiva, pois o concreto sofre o encurtamento no ato. No entanto, ao protender-se outras armaduras nessa mesma viga, a que já estava protendida sofrerá um encurtamento devido aos esforços normais adicionais aplicados sobre o concreto pelas armaduras adicionais, e assim sucessivamente a cada armadura adicionada. Esse encurtamento irá aliviar parte da tensão presente na protensão, gerando a chamada perda imediata pelo encurtamento elástico do concreto.

Caso todos os cabos fossem aplicados de uma só vez, ter-se-ia o encurtamento do concreto calculado pela seguinte expressão:

$$P_0 = n \cdot p_0 \quad (94)$$

$$\sigma_{cp} = -\frac{N_p}{A_c} - \frac{M_p}{W_{cp}} + \frac{M_g}{W_{cp}}, \text{ com } W_{cp} = \frac{I_c}{e_p} \quad (95)$$

Onde,

P_0 – força total de protensão gerada por n cabos;

n – número de cabos protendidos na peça;

p_0 – força de protensão aplicada a cada um dos n cabos individualmente;

σ_{cp} – tensão no concreto devido à protensão;

N_p – força normal devido à protensão;

M_p – momento devido à protensão;

M_g – momento devido ao peso próprio da peça;

W_{cp} – módulo de resistência à flexão a uma excentricidade e_p do CG da viga;

A_c – área de concreto da seção transversal da viga;

I_c – módulo de inércia à flexão da seção transversal do concreto;

e_p – excentricidade da protensão em relação ao centro de gravidade da seção transversal da peça.

Considerando que a protensão será realizada sobre escoramento, de modo que não haverá mobilização completa do peso próprio ou do momento de protensão, pode-se simplificar inicialmente, apenas para essa dedução, a tensão no concreto devido à protensão pela parcela apenas da força normal gerada pela protensão. Ainda, como visto no capítulo 6, aplicando a simplificação de que $N_p = P_0$ tem-se:

$$\sigma_{cp} = -\frac{N_p}{A_c} = -\frac{P_0}{A_c} \quad (96)$$

A deformação do concreto então ficaria como:

$$\sigma_{cp} = -\frac{P_0}{A_c} = \varepsilon_{cp} E_c = \frac{\Delta L_{cp}}{L} \cdot E_c \therefore \Delta L_{cp} = -\frac{P_0 L}{A_c E_c} \quad (97)$$

Onde,

ε_{cp} – deformação do concreto devido à protensão;

E_c – módulo de elasticidade do concreto;

L – comprimento da viga.

Dessa forma, pode-se escrever as seguintes expressões do encurtamento que a peça sofrerá com a protensão de cada cabo em sequência:

$$\text{cabo 1: sofrerá ação de } (n - 1) \text{ cabos após ele, logo } (n - 1) \cdot \frac{p_0 L}{A_c E_c}$$

$$\text{cabo 2: } (n - 2) \cdot \frac{p_0 L}{A_c E_c}$$

...

$$\text{cabo } (n - 1): (n - (n - 1)) \cdot \frac{p_0 L}{A_c E_c} = 1 \cdot \frac{p_0 L}{A_c E_c}$$

$$\text{cabo } n: (n - n) \cdot \frac{p_0 L}{A_c E_c} = 0$$

Onde,

n – número de cabos protendidos;

p_0 – força aplicada em cada cabo no ato da protensão.

Somando-se as parcelas acima, podemos escrever o somatório como:

$$\begin{aligned} & \sum_1^{n-1} (n - i) \frac{p_0 L}{A_c E_c} = \\ & = \left[n \sum_1^{n-1} 1 - \sum_1^{n-1} i \right] \frac{p_0 L}{A_c E_c} = \left[n(n - 1) - \frac{n(n - 1)}{2} \right] \frac{p_0 L}{A_c E_c} = \\ & = \frac{n(n - 1)}{2} \frac{p_0 L}{A_c E_c} \end{aligned} \quad (98)$$

Assim, tem-se que:

$$\Delta L_{cp} = \Delta L_p = - \frac{n(n - 1)}{2} \frac{p_0 L}{A_c E_c} = - \frac{n(n - 1)}{2} \cdot \frac{\sigma_{cp_0} L}{E_c} \quad (99)$$

O encurtamento médio para cada cabo pode ser estimado dividindo-se o ΔL_{cp} pelo número total de cabos:

$$\Delta L_{p,m} = \frac{\Delta L_{cp}}{n} = \frac{n-1}{2} \cdot \frac{\sigma_{cp_0} L}{E_c} \quad (100)$$

Como $p_{0,n} = P_0$, logo $\sigma_{p_0} = \sigma_{P_0}/n$, e então:

$$\Delta L_{p,m} = \frac{n-1}{2n} \cdot \frac{\sigma_{cp_0} L}{E_c} \quad (101)$$

Aplicando então a lei de Hooke para as expressões obtidas, da variação de comprimento médio que a protensão irá sofrer, pode-se chegar a variação de tensão causada, ou seja, a perda imediata por encurtamento elástico do concreto:

$$\Delta \sigma_p = \frac{\Delta L_{p,m} E_p}{L} = \frac{E_p}{E_c} \cdot \sigma_{cp_0} \cdot \frac{n-1}{2n} = \alpha_p \sigma_{cp_0} \cdot \frac{n-1}{2n} \quad (102)$$

Para dedução dessa fórmula desconsiderou-se o peso próprio da viga, de modo que a expressão final dada pela NBR 6118 (2014), a ser utilizada em uma futura implementação do programa será:

$$\Delta \sigma_p = \frac{\alpha_p (\sigma_{cp} + \sigma_{cg}) (n-1)}{2n} \quad (103)$$

Onde,

$\Delta \sigma_p$ – perda imediata da força de protensão por encurtamento imediato do concreto;

α_p – fator de homogeneização, relação entre os módulos de elasticidade da protensão e do concreto;

σ_{cp} – tensão de protensão no baricentro da armadura de protensão, devida aos n cabos;

σ_{cg} – tensão devido à carga permanente mobilizada simultaneamente à aplicação da protensão, no baricentro da armadura de protensão;

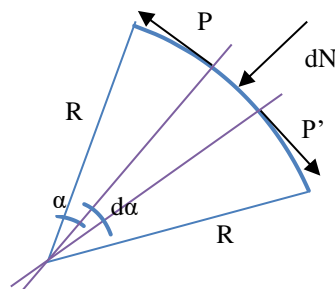
n – número de cabos protendidos na estrutura.

8.1.2 Perda Imediata por Atrito

O cabo de protensão é sinuoso, pois compõe-se de inúmeras parábolas, conforme visto no capítulo 4 anteriormente. Devido a sua forma geométrica, no ato da protensão a armadura em questão irá roçar contra a bainha, na qual está inserida, e mesmo entre os fios que a compõem, gerando forças de atrito ao longo da peça, dependente de cada posição em função de sua curvatura.

A força P do cabo sofrerá mudança de direção em seu trajeto, de modo que essa variação produzirá atrito:

Figura 12 – Força de atrito em cabo curvo.



(fonte: elaborada pelo autor.)

$$P' = P - dP \quad (104)$$

$$dP = \mu dN \quad (105)$$

Como a variação do ângulo, $d\alpha$, é muito pequena, sua tangente será aproximadamente seu próprio valor, de modo que se pode assumir que:

$$dN = P \cdot d\alpha \quad (106)$$

$$dP = -\mu \cdot P \cdot d\alpha \quad (107)$$

$$\frac{dP}{P} = -\mu \cdot d\alpha \quad (108)$$

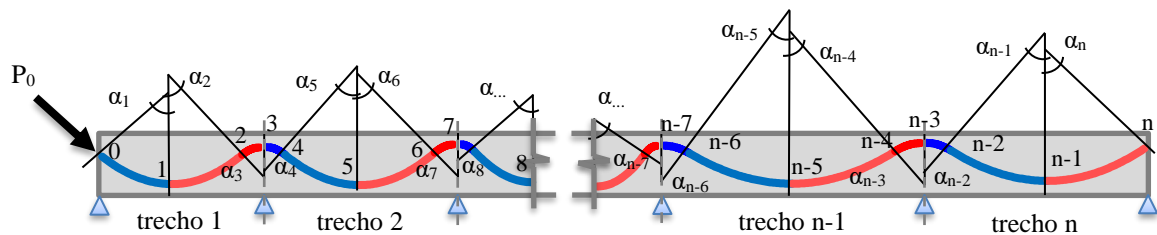
$$\ln(P) = -\mu\alpha + \ln(C) \quad (109)$$

$$\ln\left(\frac{P}{C}\right) = -\mu\alpha \quad (110)$$

$$P = C \cdot e^{-\mu\alpha} \quad (111)$$

Assim, realizando a análise de uma viga contínua com suas diversas curvas parabólicas, pode-se determinar que:

Figura 13: Variação angular do cabo de protensão ao longo de uma viga.



(fonte: elaborada pelo autor.)

$$\operatorname{tg} \alpha \cong \alpha(x) = \frac{d(y(x))}{dx} = y'(x) \quad (112)$$

$$y'(x) = \frac{d(ax^2 + bx + c)}{dx} = 2ax + b = \alpha(x) \quad (113)$$

$$\frac{d\alpha}{dx} = 2a \quad (114)$$

$$\alpha_i = \int_0^{l_i} 2a \cdot dx = 2al_i \quad (115)$$

Para as equações demonstradas anteriormente, de y_{si} , y_{ii} , y_{ij} e y_{sj} , $-a_i = a_j = f/l^2$ da respectiva parábola. Assim, os ângulos da figura acima serão:

$$\alpha_i = 2 \frac{f_i}{l_i} \quad (116)$$

$$P_0 = C \cdot e^{-\mu \cdot 0} = C \quad \therefore C = P_0 \quad (117)$$

$$P_1 = P_0 \cdot e^{-\mu \cdot \alpha_1} \quad (118)$$

$$P_2 = P_1 \cdot e^{-\mu \cdot \alpha_2} = P_0 \cdot e^{-\mu \cdot \alpha_1} \cdot e^{-\mu \cdot \alpha_2} = P_0 \cdot e^{-\mu(\alpha_1 + \alpha_2)} \quad (119)$$

$$P_3 = P_2 \cdot e^{-\mu \cdot \alpha_3} = P_0 \cdot e^{-\mu(\alpha_1 + \alpha_2)} \cdot e^{-\mu \cdot \alpha_3} = P_0 \cdot e^{-\mu(\alpha_1 + \alpha_2 + \alpha_3)} \quad (120)$$

$$\begin{aligned} P_4 &= P_3 \cdot e^{-\mu \cdot \alpha_4} = P_0 \cdot e^{-\mu(\alpha_1 + \alpha_2 + \alpha_3)} \cdot e^{-\mu \cdot \alpha_4} \\ &= P_0 \cdot e^{-\mu(\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4)} \end{aligned} \quad (121)$$

$$\begin{aligned} P_{n-1} &= P_{n-2} \cdot e^{-\mu \cdot \alpha_{n-1}} = P_0 \cdot e^{-\mu(\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \dots)} \cdot e^{-\mu \cdot \alpha_{n-1}} \\ &= P_0 \cdot e^{-\mu(\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \dots + \alpha_{n-1})} \end{aligned} \quad (122)$$

$$\begin{aligned} P_n &= P_{n-1} \cdot e^{-\mu \cdot \alpha_n} = P_0 \cdot e^{-\mu(\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \dots + \alpha_{n-1})} \cdot e^{-\mu \cdot \alpha_n} \\ &= P_0 \cdot e^{-\mu(\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \dots + \alpha_{n-1} + \alpha_n)} \end{aligned} \quad (123)$$

...

Logo,

$$P_n = P_0 \cdot e^{-\mu \Sigma \alpha_i} \quad (124)$$

$$\Delta P = P_n - P_0 = P_0 \cdot e^{-\mu \Sigma \alpha_i} - P_0 = P_0(1 - e^{-\mu \Sigma \alpha_i}) \quad (125)$$

$$\Delta P = P_0(1 - e^{-\mu \Sigma \alpha_i}) \text{ ou } \Delta P = P_0 \left(1 - e^{-2\mu \Sigma \frac{f_i}{l_i}} \right) \quad (126)$$

Ainda, além das curvaturas das parábolas, a geometria do cabo em si pode ter desvios, no caso das sinuosidades demonstradas anteriormente não seguirem uma reta, mas sim uma curva. A norma brasileira considera então um fator k multiplicado pela abscissa no expoente, representando um atrito parasita, conforme expressão abaixo:

$$\Delta P = P_i(1 - e^{-(\mu \Sigma \alpha_i + kx)}) \quad (127)$$

Onde,

ΔP – valor de perda de protensão imediata por atrito ao longo do cabo;

P_i – força de aplicação da protensão na extremidade da viga;

μ – coeficiente de atrito aparente entre cabo e bainha, estimado por:

$\mu = 0,50$ entre cabo e concreto (sem bainha);

$\mu = 0,30$ entre barras ou fios com mossas ou saliências e bainha metálica;

$\mu = 0,20$ entre fios lisos ou cordoalhas e bainha metálica;

$\mu = 0,10$ entre fios lisos ou cordoalhas e bainha metálica lubrificada;

$\mu = 0,05$ entre cordoalha e bainha de polipropileno lubrificada;

$\Sigma\alpha$ – soma dos ângulos de desvio entre o ponto de abcissa, em radianos;

k – coeficiente de perda por metro provocada por curvaturas não intencionais do cabo, que na falta de dados experimentais pode ser adotado como $0,01 \mu$ (1/m);

x – é a abcissa do ponto em que se calcula ΔP , medido a partir da ancoragem.

No caso de uma implementação no programa, deverá ter-se valores padrão para escolha do μ e um k , bem como a possibilidade para o usuário fornecer seu próprio valor para cálculo das perdas.

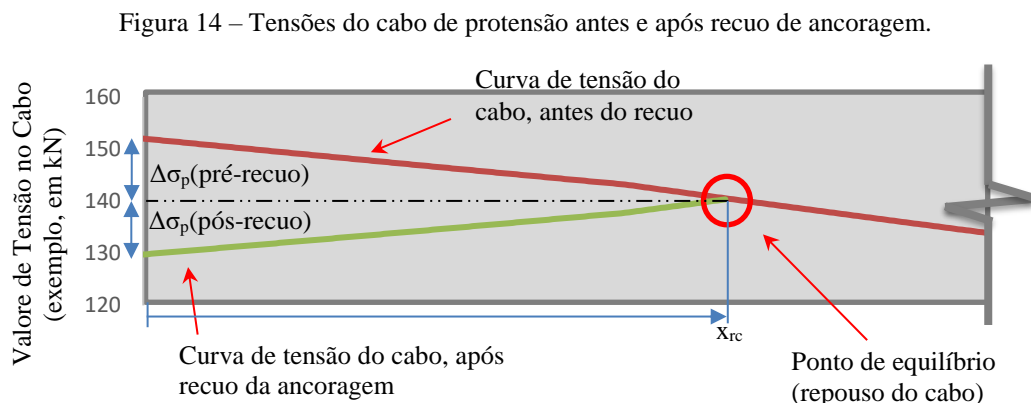
8.1.3 Perdas Por Recuo de Ancoragem

Os sistemas de aplicação da protensão nos cabos da viga podem, após executar a protensão, gerar perdas na transferência desses carregamentos para a peça. A forma de aplicação utilizada envolve diferentes métodos, sendo um dos mais comuns o com uso de cunhas para fixação do cabo em sua posição durante a protensão. No entanto, ao final de aplicação da força para então remoção do equipamento, as armaduras podem sofrer um deslizamento de sua posição ou

cravação dessas cunhas podem causar acomodação da armadura, causando perdas em sua tensão, reduzindo a força no cabo.

Devido à natureza dessas perdas e sua relação de dependência com a execução e método utilizado, os valores das mesmas podem ser determinados experimentalmente, ou determinados por meio de tabelas em função do dispositivo de ancoragem utilizado. Dessa forma, o usuário do programa deverá informar o valor do recuo de ancoragem do projeto para que sejam calculadas essas perdas, conforme segue.

Para computar essas perdas, primeiramente tem-se o valor de aplicação da protensão e o valor do recuo de ancoragem informado. O recuo significa que o cabo irá ser arrastado pelo restante preso dentro da viga, até um valor de tensão em que o atrito não mais permitirá seu arrasto. A figura abaixo ilustra essa equalização de tensões:



(fonte: elaborada pelo autor.)

Para demonstração nesse trabalho, trabalhar-se-á apenas com uma curvatura constante para maior clareza do raciocínio. O recuo da ancoragem pode ser relacionado com a deformação do cabo, pela lei de Hooke, conforme segue:

$$\varepsilon^{\lambda_a}(x) = \frac{\Delta\sigma_p^{\lambda_a}(x)}{E_p} \quad (128)$$

$$\lambda_a = \int_0^{x_{rc}} \varepsilon^{\lambda_a}(x) \cdot dx \quad (129)$$

$$\lambda_a = \int_0^{x_{rc}} \frac{\Delta\sigma_p^{\lambda_a}(x)}{E_p} \cdot dx \quad (130)$$

Onde,

ε^{λ_a} – deformação no cabo devida ao recuo da ancoragem;

$\Delta\sigma_p^{\lambda_a}$ – variação da tensão no cabo entre antes e depois do recuo da ancoragem;

E_p – módulo de deformação da armadura de protensão;

λ_a – valor do recuo da ancoragem (m);

x_{rc} – abscissa do ponto de repouso do cabo, isto é, onde a tensão do cabo com as perdas por atrito se iguala à tensão no cabo com as perdas por recuo de ancoragem.

Com uso de algumas adaptações das equações demonstradas na seção 9.1.2, pode-se determinar a equação para a variação da tensão do cabo em um ponto x distante de sua origem.

$$\Delta\sigma_p^{\lambda_a}(x) = \sigma_{p1}^{\mu}(x) - \sigma_{p2}^{\mu}(x) \quad (131)$$

$$\sigma_{p1}^{\mu}(x) = \sigma_{p0} \cdot e^{-\mu(\alpha+\gamma)x}, \text{ antes do recuo} \quad (132)$$

$$\sigma_{p2}^{\mu}(x) = \sigma_{p1}^{\mu}(x_{rc}) \cdot e^{-\mu(\alpha+\gamma)(x_{rc}-x)}, \text{ após recuo} \quad (133)$$

$$\Delta\sigma_p^{\lambda_a}(x) = \sigma_{p0} \cdot e^{-\mu(\alpha+\gamma)x} - \sigma_{p1}^{\mu}(x_{rc}) \cdot e^{-\mu(\alpha+\gamma)(x_{rc}-x)} \quad (134)$$

$$\Delta\sigma_p^{\lambda_a}(x) = \sigma_{p0} \cdot e^{-\mu(\alpha+\gamma)x} - \sigma_{p0} \cdot e^{-\mu(\alpha+\gamma)x_{rc}} \cdot e^{-\mu(\alpha+\gamma)(x_{rc}-x)} \quad (135)$$

$$\Delta\sigma_p^{\lambda_a}(x) = \sigma_{p0} \cdot (e^{-\mu(\alpha+\gamma)x} - e^{-\mu(\alpha+\gamma)(2x_{rc}-x)}) \quad (136)$$

Substituindo-se a equação (163) na equação (157), pode-se encontrar a equação geral para cálculo do recuo da ancoragem:

$$\lambda_a = \int_0^{x_{rc}} \frac{\sigma_{p0} \cdot (e^{-\mu(\alpha+\gamma)x} - e^{-\mu(\alpha+\gamma)(2x_{rc}-x)})}{E_p} \cdot dx \quad (137)$$

$$\lambda_a = \frac{\sigma_{p0}}{E_p} \int_0^{x_{rc}} (e^{-\mu(\alpha+\gamma)x} - e^{-\mu(\alpha+\gamma)(2x_{rc}-x)}) \cdot dx \quad (138)$$

$$\lambda_a = \left| -\frac{\sigma_{p0} (e^{-\mu(\alpha+\gamma)x} + e^{-\mu(\alpha+\gamma)(2x_{rc}-x)})}{E_p \mu(\alpha+\gamma)} + c \right|_0^{x_{rc}} \quad (139)$$

$$\lambda_a = \frac{\sigma_{p0}}{E_p} \left[-\frac{(e^{-\mu(\alpha+\gamma)x_{rc}} + e^{-\mu(\alpha+\gamma)x_{rc}})}{\mu(\alpha+\gamma)} + \frac{(e^0 + e^{-\mu(\alpha+\gamma)2x_{rc}})}{\mu(\alpha+\gamma)} \right] \quad (140)$$

$$\lambda_a = \frac{\sigma_{p0}}{E_p} \frac{e^{-\mu(\alpha+\gamma)2x_{rc}} - 2 \cdot e^{-\mu(\alpha+\gamma)x_{rc}} + 1}{\mu(\alpha+\gamma)} \quad (141)$$

$$\lambda_a = \frac{\sigma_{p0}}{E_p} \frac{(e^{-\mu(\alpha+\gamma)x_{rc}} - 1)^2}{\mu(\alpha+\gamma)} \quad (142)$$

$$\frac{\lambda_a E_p}{\sigma_{p0}} \mu(\alpha+\gamma) = (e^{-\mu(\alpha+\gamma)x_{rc}} - 1)^2 \quad (143)$$

$$1 \pm \sqrt{\frac{\lambda_a E_p}{\sigma_{p0}} \mu(\alpha+\gamma)} = e^{-\mu(\alpha+\gamma)x_{rc}} \quad (144)$$

$$\ln \left(1 \pm \sqrt{\frac{\lambda_a E_p}{\sigma_{p0}} \mu(\alpha+\gamma)} \right) = \ln(e^{-\mu(\alpha+\gamma)x_{rc}}) \quad (145)$$

$$x_{rc} = \frac{\ln \left(1 \pm \sqrt{\frac{\lambda_a E_p}{\sigma_{p0}} \mu(\alpha+\gamma)} \right)}{-\mu(\alpha+\gamma)} \quad (146)$$

Com base nessa equação (173), é possível obter o ponto de repouso do cabo para um recuo de ancoragem informado pelo projetista, e então calcular as perdas imediatas adicionais decorrentes.

8.2 PERDAS PROGRESSIVAS

As perdas progressivas estão relacionadas à passagem do tempo, decorrentes de fenômenos reológicos característicos do concreto e do aço da protensão. O concreto gera perdas por retração e pela fluência, ao passo que a armadura gera perdas devido à relaxação do aço.

8.2.1 Relaxação do Aço

A relaxação do aço é dependente do tipo do material, da tensão aplicada no aço, da temperatura e do tempo decorrido desde sua tração inicial em um tempo inicial t_0 até um instante t em que se deseje determinar o valor da perda. Pela NBR 6118 (2014) pode-se determinar essa perda de maneira direta, pelos seguintes passos:

$$\Delta\sigma_{pr}(t, t_0) = \psi(t, t_0) \cdot \sigma_{pi} \quad (147)$$

$$\psi(t, t_0) = \psi_{1000} \left(\frac{t - t_0}{41,67} \right)^{0,15} \quad (148)$$

Onde,

$\Delta\sigma_{pr}(t, t_0)$ – perda de tensão por relaxação pura desde o instante t_0 do estiramento da armadura até o instante t considerado;

$\psi(t, t_0)$ – intensidade da relaxação do aço;

ψ_{1000} – valores médios de relaxação medidos após 1000 h à temperatura constante de 20 °C, definidos na Tabela 4;

σ_{pi} – tensão inicial aplicada na armadura de protensão;

t e t_0 – idades final e inicial do concreto para o intervalo considerado (dias).

É possível ainda substituir-se a equação (148) para entrada de t e t_0 em horas, resultando em:

$$\psi(t, t_0) = \psi_{1000} \left(\frac{t - t_0}{1000} \right)^{0,15} \quad (149)$$

O valor médio de relaxação ψ_{1000} é fornecido na Tabela 4, considerando-se que para σ_{pi} inferior à 50% de f_{ptk} do aço não existe perda por relaxação. A Tabela 4 apresenta valores para 50% e 80%, sendo que os demais dentro dessa faixa são obtidos por interpolação. Ainda, vale observar que para períodos de tempo que seguem ao infinito, o valor de $\psi(t, t_0)$ pode ser aproximado para $\psi(t_{\infty}, t_0) \approx 2,5 \psi_{1000}$.

Tabela 4 – Valores de ψ_{1000} , em porcentagem

σ_{p0}	Cordoalhas		Fios		Barras
	RN	RB	RN	RB	
0,5 f_{ptk}	0	0	0	0	0
0,6 f_{ptk}	3,5	1,3	2,5	1,0	1,5
0,7 f_{ptk}	7,0	2,5	5,0	2,0	4,0
0,8 f_{ptk}	12,0	3,5	8,5	3,0	7,0

Onde
 RN é a relaxação normal;
 RB é a relaxação baixa.

(fonte: adaptada pelo autor da NBR 6118:2014.)

8.2.2 Deformações do Concreto no Tempo

O concreto sofrerá deformações no tempo devido à presença de carregamentos permanentes, incluindo aqueles gerados pela armadura de protensão. As deformações consideradas aqui serão as causadas pelo fenômeno de fluência do concreto, bem como aquela por retração no tempo. Essas perdas associadas com a deformação imediata do concreto, resultarão na deformação total computada para a viga em si. A expressão pela NBR 6118 (2014) é:

$$\varepsilon_c(t) = \varepsilon_c(t_0) + \varepsilon_{cc}(t) + \varepsilon_{cs}(t) \quad (150)$$

Onde,

$\varepsilon_c(t)$ – deformação do concreto devido a uma tensão constante em um intervalo de tempo de t_0 a t ;

$\varepsilon_c(t_0)$ – deformação imediata do concreto por ocasião do carregamento;

$\varepsilon_{cc}(t)$ – deformação por fluência do concreto, no intervalo de tempo de t_0 a t ;

$\varepsilon_{cs}(t)$ – deformação por retração do concreto, no intervalo de tempo de t_0 a t .

Apesar de não haver aderência entre a armadura de protensão e o concreto no caso da pós-tração, a hipótese de que as deformações do concreto e do aço são as mesmas é admitida pela norma.

8.2.2.1 Fluência do Concreto

A fluência do concreto divide-se em três componentes, sendo uma relativa a uma deformação rápida e as outras duas a uma deformação lenta, uma reversível e a outra irreversível, conforme expressão da NBR 6118 (2014):

$$\varepsilon_{cc}(t, t_0) = \varepsilon_{cca} + \varepsilon_{ccf} + \varepsilon_{ccd} = \frac{\sigma_c}{E_{c28}} \varphi(t, t_0) \quad (151)$$

Onde,

ε_{cca} – deformação rápida irreversível, referente às primeiras 24 h após a aplicação da carga que a origina;

ε_{ccf} – deformação lenta irreversível;

ε_{ccd} – deformação lenta reversível;

σ_c – tensão no concreto devido à ação das cargas permanentes, no intervalo de tempo de t_0 a t ;

E_{c28} – módulo de deformação tangente inicial para $j = 28$ dias;

$\varphi(t, t_0)$ – coeficiente de fluência para o concreto.

O coeficiente de fluência do concreto é dado na NBR 6118 (2014) por:

$$\varphi(t, t_0) = \varphi_a + \varphi_{f\infty}[\beta_f(t) - \beta_f(t_0)] + \varphi_{d\infty}\beta_d \quad (152)$$

Onde,

t – idade fictícia do concreto no instante considerado (dias);

t_0 – idade fictícia do concreto ao ser feito um carregamento único (dias);

φ_a – coeficiente de fluência rápida;

$\varphi_{f\infty}$ - coeficiente de deformação lenta irreversível;

$\varphi_{d\infty}$ - valor do coeficiente de deformação lenta reversível, que é considerado igual a 0,4;

$\beta_f(t)$ ou $\beta_f(t_0)$ - coeficiente de deformação lenta irreversível em função da idade do concreto;

$\beta_d(t)$ – é o coeficiente relativo à deformação lenta reversível do tempo $(t - t_0)$ decorrido após o carregamento.

O coeficiente de fluência rápida é determinado, conforme NBR 6118 (2014), por:

$$\varphi_a = 0,8 \left[1 - \frac{f_c(t_0)}{f_c(t_\infty)} \right], \text{ para concretos de C20 a C45.} \quad (153)$$

$$\varphi_a = 1,4 \left[1 - \frac{f_c(t_0)}{f_c(t_\infty)} \right], \text{ para concretos de C50 a C90.} \quad (154)$$

Onde,

$f_c(t)/f_c(t_\infty)$ – é a função do crescimento da resistência do concreto com a idade, determinada conforme capítulo 5.1;

O coeficiente de deformação lenta irreversível $\varphi_{f\infty}$ é determinado, pela NBR 6118 (2014), por:

$$\varphi_{f_{\infty}} = \varphi_{1c} \cdot \varphi_{2c}, \text{ para concretos de C20 a C45.} \quad (155)$$

$$\varphi_{f_{\infty}} = 0,45 \cdot \varphi_{1c} \cdot \varphi_{2c}, \text{ para concretos de C50 a C90.} \quad (156)$$

O coeficiente φ_{1c} é obtido pela Tabela 5 que segue:

Tabela 5 – Valores numéricos usuais para determinação da fluência e retração.

Ambiente	Umidade U %	Fluência $\varphi_{1c}^{a,c}$			Retração $10^4 \varepsilon_{1s}^{b,c}$			γ_d
		Abatimento de acordo com a ABNT NBR NM 67 cm						
		0 – 4	5 – 9	10 – 15	0 – 4	5 – 9	10 – 15	
Na água	–	0,6	0,8	1,0	+1,0	+1,0	+1,0	30,0
Em ambiente muito úmido imediatamente acima da água	90	1,0	1,3	1,6	– 1,9	– 2,5	– 3,1	5,0
Ao ar livre, em geral	70	1,5	2,0	2,5	– 3,8	– 5,0	– 6,2	1,5
Em ambiente seco	40	2,3	3,0	3,8	– 4,7	– 6,3	– 7,9	1,0

^a $\varphi_{1c} = 4,45 - 0,035U$ para abatimento no intervalo de 5 cm a 9 cm e $U \leq 90\%$.

^b $10^4 \varepsilon_{1s} = -8,09 + (U/15) - (U^2/2.284) + (U^3/133.765) - (U^4/7.608.150)$ para abatimentos de 5 cm a 9 cm e $40\% \leq U \leq 90\%$.

^c Os valores de φ_{1c} e ε_{1s} para $U \leq 90\%$ e abatimento entre 0 cm e 4 cm são 25% menores e, para abatimentos entre 10 cm e 15 cm, são 25% maiores.

^d $\gamma = 1 + \exp(-7,8 + 0,1 U)$ para $U \leq 90\%$.

NOTA 1 Para efeito de cálculo, as mesmas expressões e os mesmos valores numéricos podem ser empregados, no caso de tração.

NOTA 2 Para o cálculo dos valores de fluência e retração, a consistência do concreto é aquela correspondente à obtida com o mesmo traço, sem adição de superplastificantes e superfluidificantes.

(fonte: adaptada pelo autor da NBR 6118:2014.)

O coeficiente φ_{2c} depende da espessura fictícia do concreto, conforme expressão da NBR 6118 (2014):

$$\varphi_{2c} = \frac{42 + h_{fic}}{20 + h_{fic}} \quad (157)$$

Onde,

h_{fic} – altura fictícia do concreto em cm.

O cálculo de h_{fic} é dado na NBR 6118 (2014) por:

$$h_{fic} = \gamma \frac{2A_c}{u_{ar}} \quad (158)$$

Onde,

γ – coeficiente dependente da umidade relativa do ambiente (U%), conforme Tabela 5,

A_c – área da seção transversal da peça;

u_{ar} – perímetro da seção transversal da peça que fica em contato com o ar.

Para determinar-se γ , conforme NBR 6118 (2014), usa-se a expressão:

$$\gamma = 1 + \exp(-7,8 + 0,1U) \quad (159)$$

Os coeficientes relativos às deformações lentas irreversível e reversível, são dados na NBR 6118 (2014) por respectivamente:

$$\beta_f(t) = \frac{t^2 + At + B}{t^2 + Ct + D} \quad (160)$$

$$\beta_d(t) = \frac{t - t_0 + 20}{t - t_0 + 70} \quad (161)$$

Onde,

$$A = 42h^3 - 350h^2 + 588h + 133 \quad (162)$$

$$B = 768h^3 - 3060h^2 + 3234h - 23 \quad (163)$$

$$C = -200h^3 + 13h^2 + 1090h + 183 \quad (164)$$

$$D = 7579h^3 - 31916h^2 + 35343h + 1931 \quad (165)$$

Onde,

h – espessura fictícia em metros (m); para valores de h fora do intervalo ($0,05 \leq h \leq 1,6$) adotam-se os extremos correspondentes;

t – tempo em dias, sendo $t \geq 3$.

8.2.2.2 Retração do Concreto

A retração do concreto será dependente de fatores como umidade relativa do ambiente, consistência do concreto no lançamento, bem como da espessura fictícia da peça. A expressão de cálculo do valor de retração é dada na NBR 6118 (2014) por:

$$\varepsilon_{cs}(t, t_0) = \varepsilon_{cs\infty} [\beta_s(t) - \beta_s(t_0)] \quad (166)$$

Sendo,

$$\varepsilon_{cs\infty} = \varepsilon_{1s} \varepsilon_{2s} \quad (167)$$

Onde,

$\varepsilon_{cs\infty}$ – valor final da retração;

ε_{1s} – coeficiente de relação da retração com a umidade relativa do ambiente e com a consistência do concreto, conforme Tabela 5;

ε_{2s} – coeficiente de relação da retração com a espessura fictícia da peça;

$\beta_s(t)$ e $\beta_s(t_0)$ – coeficiente relativo à retração no instante t ou t_0 .

A relação da retração do concreto com a espessura fictícia da peça é dada por:

$$\varepsilon_{2s} = \frac{33 + 2h_{fic}}{20,8 + 3h_{fic}} \quad (168)$$

O coeficiente de relação da retração com o tempo é dada na NBR 6118 (2014) por:

$$\beta_s(t) = \frac{\left(\frac{t}{100}\right)^3 + A\left(\frac{t}{100}\right)^2 + B\left(\frac{t}{100}\right)}{\left(\frac{t}{100}\right)^3 + C\left(\frac{t}{100}\right)^2 + D\left(\frac{t}{100}\right) + E} \quad (169)$$

Onde,

$$A = 40 \quad (170)$$

$$B = 116h^3 - 282h^2 + 220h - 4,8 \quad (171)$$

$$C = 2,5h^3 - 8,8h + 40,7 \quad (172)$$

$$D = -75h^3 + 585h^2 + 496h - 6,8 \quad (173)$$

$$E = -169h^4 + 88h^3 + 584h^2 - 39h + 0,8 \quad (174)$$

Por fim, a idade fictícia normaliza a idade do concreto com o endurecimento do cimento. A velocidade de endurecimento pode variar dependendo do tipo de cura e temperatura a que este endurece, de modo que para relacionar diferentes concretos em um único padrão utiliza-se a idade fictícia.

A idade fictícia do concreto é determinada pelas seguintes expressões, quando não houver cura a vapor, conforme NBR 6118 (2014):

$$t = \alpha t_{ef}, \text{ para cura à temperatura ambiente de } 20^\circ\text{C} \quad (175)$$

$$t = \alpha \sum_i \frac{T_i + 10}{30} \Delta t_{ef,i}, \text{ para demais casos} \quad (176)$$

Onde,

α – coeficiente que relaciona a velocidade de endurecimento do concreto, dado pela Tabela 6;

T_i – temperatura média diária do ambiente (°C);

$\Delta t_{ef,i}$ – período durante o qual a temperatura média diária do ambiente, T_i pode ser admitida constante (dias).

Tabela 6 – Valores de fluência e de retração em função da velocidade de endurecimento do cimento

Cimento Portland (CP)	α	
	Fluência	Retração
De endurecimento lento (CP III e CP IV, todas as classes de resistência)	1	1
De endurecimento normal (CP I e CP II, todas as classes de resistência)	2	
De endurecimento rápido (CP V-ARI)	3	
Legenda: CP I e CP I-S – Cimento Portland comum CP II-E, CP II-F e CP II-Z – Cimento Portland composto CP III – Cimento Portland de alto forno CP IV – Cimento Portland pozolânico CP V-ARI – Cimento Portland de alta resistência inicial RS – Cimento Portland resistente a sulfatos (propriedade específica de alguns tipos de cimento citados)		

(fonte: adaptada pelo autor da NBR 6118:2014.)

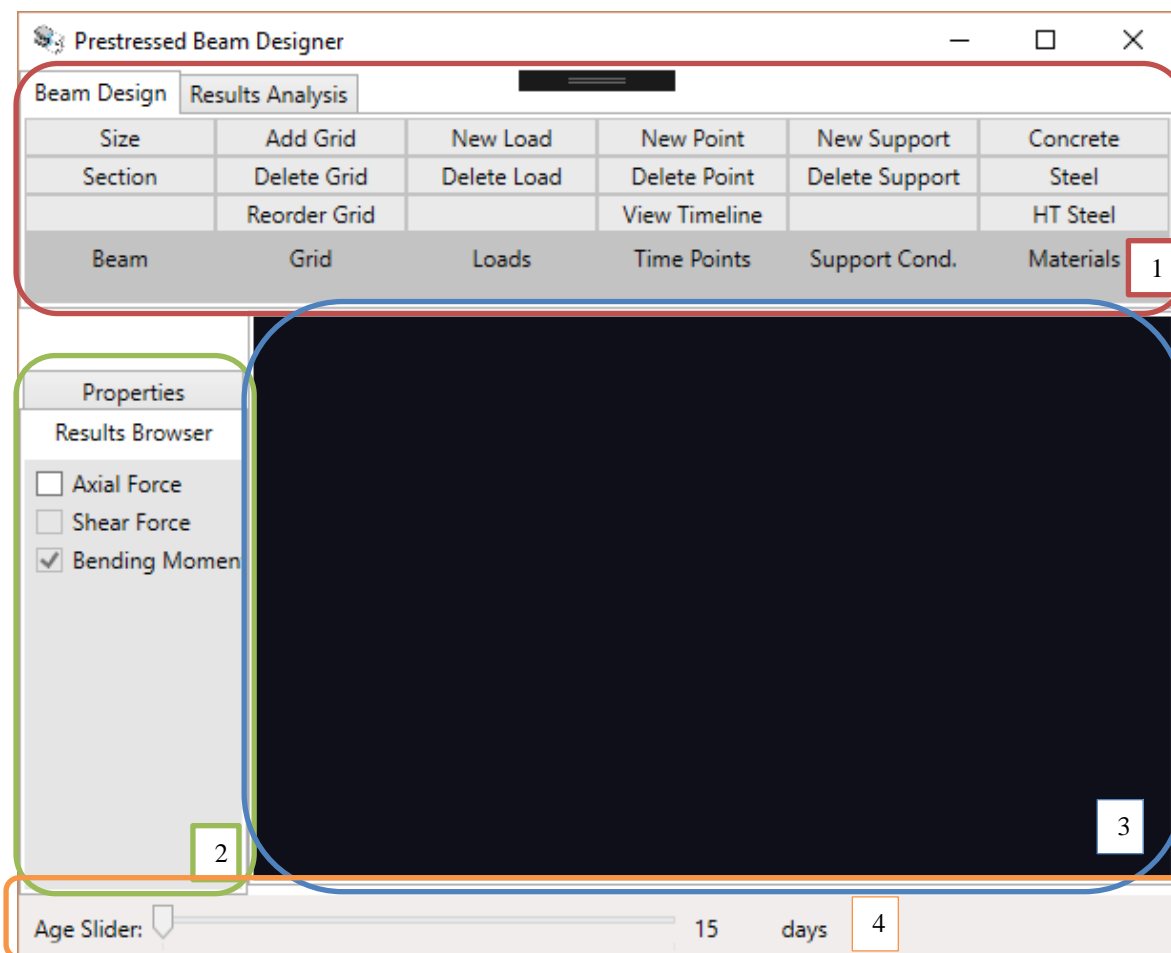
9 IMPLEMENTAÇÃO COMPUTACIONAL

A implementação computacional desse trabalho envolve duas atividades principais: uma de desenvolvimento de uma interface para coleta dos dados e parâmetros necessários de projeto e para exibição de resultados para análise, e outra de aplicação dos modelos matemáticos desenvolvidos em forma de algoritmos computacionais para processar esses dados e então devolvê-los novamente em uma interface com o usuário. Dessa forma, serão apresentados primeiramente os elementos de interface do programa e seu uso, para compreensão da entrada de dados, e após os principais algoritmos que irão tratar esses dados por trás da interface para os devolver ao usuário de maneira transparente.

9.1 ELEMENTOS DO PROGRAMA

A entrada de dados é umas das principais etapas do programa pois neste ponto o usuário define seu projeto de maneira geral, sendo essencial o entendimento de como cada informação fornecida será interpretada pelo *software* desenvolvido para se obter os resultados esperados com confiabilidade. O ponto de vista crítico do engenheiro é essencial para validação dos resultados obtidos.

Figura 15 – Interface principal do programa.



(fonte: elaborada pelo autor.)

A Figura 15 ilustra a interface principal do programa, separando cada um dos seus elementos, conforme indicação numérica que segue: barra de ferramentas composta por botões de comandos (1), barra de controles com seletor de elementos de visualização (2), barra de seleção de idade do concreto por meio de marcador e área interativa de visualização (4) e entrada de dados por meio gráfico em duas e três dimensões (3). Os comandos, por sua vez, podem criar novos elementos para coleta de dados, como caixas de texto, janelas de diálogo com formulários (Figura 16) ou por interação direta na área interativa com uso de *mouse*

Figura 16 – Exemplo de janela de diálogo com formulário para coleta de dados.

The dialog box is titled "Support Conditions" and features a close button (X) in the top right corner. It is organized into three main sections for different types of support conditions:

- Displacement - X direction:** Includes a checkbox for "Fixed", a "Spring" input field with the value "0" and unit "kN/m", and a "Prescribed" input field with the value "0" and unit "mm".
- Displacement - Y direction:** Includes a checkbox for "Fixed", a "Spring" input field with the value "0" and unit "kN/m", and a "Prescribed" input field with the value "0" and unit "mm".
- Rotation - Z direction:** Includes a checkbox for "Fixed", a "Spring" input field with the value "0" and unit "kN.m/rad", and a "Prescribed" input field with the value "0" and unit "rad".

To the right of these sections is a "Preview" window showing a black square with a red dot in the center, representing the support location. Below the preview are three buttons: "Insert", "Reset", and "Cancel".

(fonte: elaborada pelo autor.)

9.1.1 Geometria

A entrada de dados é feita pelo uso de objetos de programação, representados graficamente, por divisões de partes da estrutura em viga, seção, restrições, cargas, marcas de tempo e *grid*. Os detalhes sobre dados vinculados a cada objeto serão melhor descritos por elemento nas seções seguintes.

9.1.1.1 Seção

A determinação da seção pelo usuário pode ser realizada por meio de duas etapas. Primeiro, deve ser escolhido o tipo da seção a partir de uma biblioteca de tipos, sendo inicialmente disponíveis os formatos: retangular e I. Segundo, as dimensões da seção escolhida podem ser alteradas no programa para os valores desejados. Todas as propriedades da seção são calculadas em tempo real para posterior uso nos demais cálculos de projeto. Os parâmetros obtidos nessa etapa são: momento de inércia de área, centro geométrico, perímetro e área.

9.1.1.2 Viga

A viga permite acesso para alterar o comprimento da mesma, bem como para acessar os grids, cargas e restrições sobre a estrutura. As solicitações ao longo da viga serão calculadas com base nesses dados, uma vez sejam solicitadas as informações sobre elas.

Para alterar o tamanho da viga utiliza-se o botão “Size” da barra “Beam Design” que irá abrir um campo de texto com o tamanho atual da viga. Através do teclado é possível alterar esse valor, validando-o por meio da tecla “Enter” que irá então atualizar a representação gráfica da viga, bem como irá alertar ao usuário possíveis modificações que a operação poderá causar nas informações previamente fornecidas como remoção de grids, cargas ou apoios no caso de diminuição do tamanho do elemento.

9.1.1.3 Grids

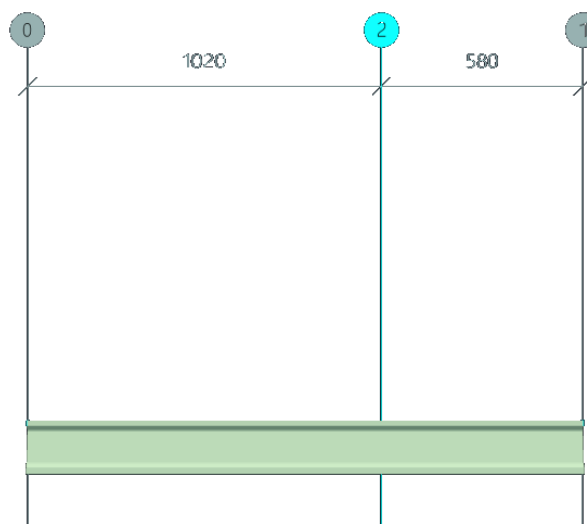
Esses elementos destinam-se a auxiliar o posicionamento de cargas e restrições ao longo do corpo da viga, bem como servem de informação para a criação de nós e segmentação da viga para geração das matrizes destinadas à resolução da estrutura.

A chamada da função para inserir grids se dá pelo botão “Add Grid” na aba “Beam Design”, que então lê as informações do *mouse* na área interativa para posicionar e desenhar um novo grid ao longo do comprimento da viga previamente definido. O posicionamento durante a inserção possibilita uma precisão de 10 em 10 cm. A posição é ilustrada em centímetros por meio de uma régua entre cada *grid* exibida graficamente. Para precisões maiores é possível, após inserido o *grid*, girar o botão de *scroll* do *mouse* que irá realizar deslocamentos a cada 0,5 cm. Ainda, para ajustar *grids* já inseridos é possível também arrastá-los selecionando-os por um clique do botão esquerdo do *mouse*, de modo que se mantenha esse botão pressionado ao movimentar o *mouse* que irá reposicionar o grid ao longo da viga. Considerou-se que 0,5 cm na prática é suficiente para o projeto de vigas visto que não é prático o uso medidas menores do que essa em sua execução.

Para implementações futuras de melhor da interface seria possível implementar também a entrada da posição desses *grids* por meio do teclado, lendo distâncias em relação a alguma referência e então atualizando a posição por uso das rotinas já desenvolvidas. A Figura 17 ilustra a representação gráfica dos *grids* ao longo da viga, e como estes são representados durante a

inserção de um novo elemento (o fundo escuro da imagem foi removido para otimizar a impressão desse documento).

Figura 17 – Inserção de novo elemento *grid* ao longo da viga.



(fonte: elaborada pelo autor.)

Outra função, como a renumeração dos grids, foi parcialmente implementada para fins de demonstração. O reordenamento dos *grids* é acessível pelo botão “Reorder Grid” que não necessita interação do usuário, refazendo suas estruturas internas, conexões destes com cargas e com apoios, atualizando sua visualização gráfica.

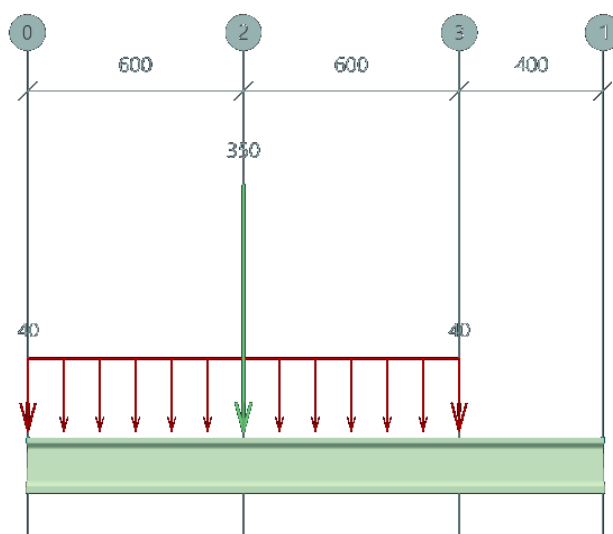
9.1.2 Cargas

A entrada de cargas dá-se por meio de etapas, com a seleção da função de carga (força pontual vertical, momento pontual, distribuída constante, distribuída triangular), com a definição da magnitude da carga, com a seleção do período de tempo em que ela atua (pelo meio de marcas temporais), e com o posicionamento da carga por meio dos pontos de *grid* existentes. Cada *grid* então, por sua vez, receberá parte das cargas inseridas de maneira equivalente a uma única carga concentrada em seu ponto, para posterior uso nas matrizes que computam a estrutura.

Cargas móveis não serão incluídas nessa primeira versão do programa, ficando para implementação futura entre outros melhoramentos.

O botão “New Load” da barra de comandos “Beam Design” exibe em uma nova janela de diálogo todos os campos necessários para definição dos parâmetros das cargas, como tipo de carga e valores. Após a definição desses parâmetros, estes são então representados graficamente na área interativa da interface para que sejam selecionados os *grids* sobre os quais o usuário desejaocar a carga. Por exemplo, para inserção de uma carga distribuída, o usuário necessitará escolher dois *grids* para sinalizar o início e o fim do trecho em que a carga irá atuar, utilizando o *mouse* e realizando cliques com o botão esquerdo do mesmo. Antes ou durante a inserção da carga, é possível selecionar a partir de que idade do concreto deseja-se que essa atue, arrastando o seletor de idades do concreto na janela principal, ou realizando cliques com o botão esquerdo do mouse ao longo deste, e então continuar inserindo a carga. A Figura 18 a seguir ilustra a representação de duas cargas inseridas sobre a viga, com exibição dos *grids* que definem sua localização.

Figura 18 – Representação de cargas ao longo da viga.



(fonte: elaborada pelo autor.)

9.1.3 Restrições

As restrições são aplicadas pelo meio de molas (rigidezes equivalentes), vínculos perfeitos ou deslocamentos definidos, que representem as condições de apoio e engastamento da viga. Assim como na inserção de cargas, as restrições são em três etapas: seleção das direções restringidas (horizontal, vertical ou circular), inserção do valor da restrição, e posicionamento da mesma em algum ponto do *grid* previamente definido pelo projetista. Nas rotinas para resolução da estrutura, esses valores de rigidez irão somar na matriz de rigidez das barras, alterando assim, o valor naqueles nós, computando-se sua contribuição nos resultados das solicitações.

O comando para inserir uma nova restrição é feito pelo botão “New Support” da barra “Beam Design” que irá abrir uma janela de diálogo com campos para definição da carga, bem como uma área de ilustração dos dados inseridos, a medida que esses são digitados. O botão “Insert” dessa janela levará o usuário então à área interativa para posicionamento dos apoios na viga, o botão “Reset” reinicializa os dados da janela, e o botão “Cancel” aborta a operação.

A Figura 19 a seguir demonstra a representação de previsão (“Preview”) conforme parâmetros são selecionados para o apoio a ser inserido, bem como alguns exemplos da sua representação final já inserido na viga (Figura 20).

Figura 19 - Representação gráfica das restrições conforme parâmetros selecionados.

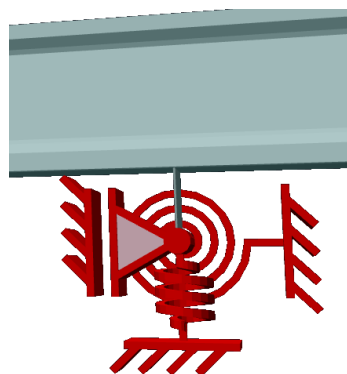
The screenshot shows a dialog box titled "Support Conditions" with a close button (X) in the top right corner. It contains three sections for defining support parameters:

- Displacement - X direction:** The "Fixed" checkbox is checked. The "Spring" field contains the value "0" with units "kN/m". The "Prescribed" field contains the value "0" with units "mm".
- Displacement - Y direction:** The "Fixed" checkbox is unchecked. The "Spring" field contains the value "150000" with units "kN/m". The "Prescribed" field contains the value "0" with units "mm".
- Rotation - Z direction:** The "Fixed" checkbox is unchecked. The "Spring" field contains the value "25000000" with units "kN.m/rad". The "Prescribed" field contains the value "0" with units "rad".

To the right of these sections is a "Preview" window showing a red support symbol on a black background. Below the preview are three buttons: "Insert", "Reset", and "Cancel".

(fonte: elaborada pelo autor.)

Figura 20 - Representação gráfica das restrições conforme parâmetros selecionados.



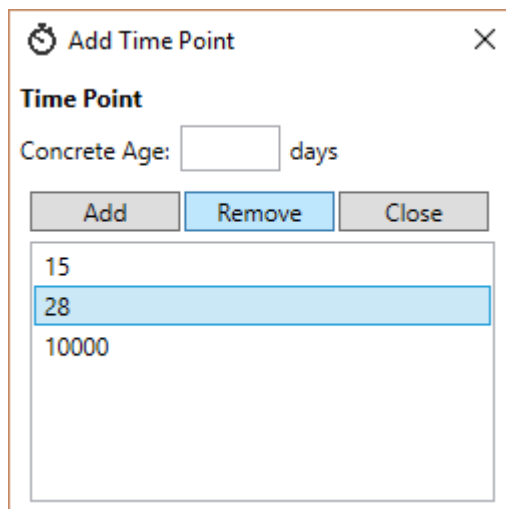
(fonte: elaborada pelo autor.)

9.1.4 Instantes temporais

Os instantes temporais possibilitam ao usuário definir quantas idades julgar necessário para o concreto, gerando uma lista na qual poderá então relacionar condições diversas de carregamentos para a viga. Com essa estrutura relacionada de carregamentos e idades do concreto, a estrutura de dados formada possibilita que em uma continuidade de desenvolvimento do programa seja possível calcular as perdas temporais para o concreto e para o aço de protensão, baseado no período de atuação de cada carga inserida.

O programa inicia com dois instantes de tempo já pré-definidos, nas idades de 15 dias e infinito (10.000 dias) do concreto. Para alterar essa lista, é necessário clicar no botão “New Point” da barra “Beam Design” que irá abrir então as funções necessárias em uma nova caixa de diálogo. Essa nova caixa possui botões de “Add”, para adicionar novas idades, informadas por meio do teclado na caixa de texto “Concrete Age:”, “Remove”, para remover alguma idade por meio de seleção da idade que se deseja remover de lista exibida na própria janela (Figura 21), e “Close” para encerrar o comando.

Figura 21 - Remoção de um instante de tempo da lista de idades do concreto.



(fonte: elaborada pelo autor.)

9.1.5 Materiais

A definição dos parâmetros dos materiais serve à definição de variáveis necessárias ao cálculo de diferentes equações de projeto. A partir destes, o programa realiza diversas definições internas que possibilitam a implementação futura de novas funções e equações com esses dados já coletados.

Existem três botões na barra de “Design Beam” para definição dos materiais, para facilidade de acesso, no entanto que chamam uma mesma janela de diálogo de materiais, organizados em abas separadas. A escolha de cada um dos botões serve para facilitar o acesso de modo que a janela ao abrir já esteja com aba desejada selecionada.

O botão “Concrete” abre o diálogo que possibilita entrar com os parâmetros de classe de compressão do concreto, tipo de concreto, tipo de agregado, condições de verificação e classe de agressividade ambiental, por meio de seleção de listas expansíveis já pré-definidas internamente no programa, conforme classificações existentes na NBR 6118:2014. Apenas a umidade relativa do ar é informada por uma caixa de texto por uso do teclado, em %.

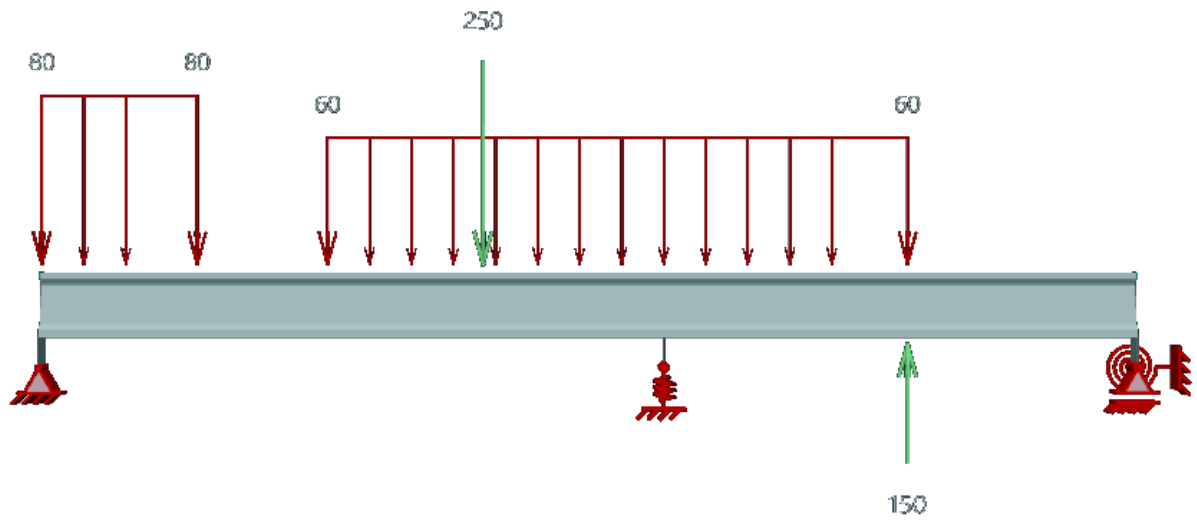
Os demais materiais poderão ser definidos de igual modo em futura implementação.

9.1.6 Resultados da Estrutura

Após a entrada dos demais elementos mencionados nesse capítulo, é possível obter-se as solicitações da estrutura definida. Os resultados são computados no mesmo instante em que sua exibição é solicitada, por meio do botão “Results Plot” na aba “Results Analysis” da barra de ferramentas. O programa realiza internamente todas as operações necessárias de maneira transparente: definindo módulo de elasticidade do concreto, área e momento de inércia de área da seção transversal da viga; cálculo do comprimento de cada trecho da viga entre apoios, cálculo da matriz de rigidez para cada trecho e da matriz de rigidez para a viga completa, cálculo das cargas nodais pelo método dos deslocamentos; aplicação das restrições e simplificações da matriz de rigidez e do vetor de forças nodais (conforme explicado em detalhes no Capítulo 3); solução do sistema de equações pelo método de eliminação de Gauss, obtendo os deslocamentos nodais; cálculo das reações em cada nó, cálculo das solicitações ao longo das barras com uso das reações e cargas aplicadas em cada trecho; geração dos elementos de dados gráficos para representação dos dados, finalizando com a exibição destes na tela interativa, alterando para modo bidimensional.

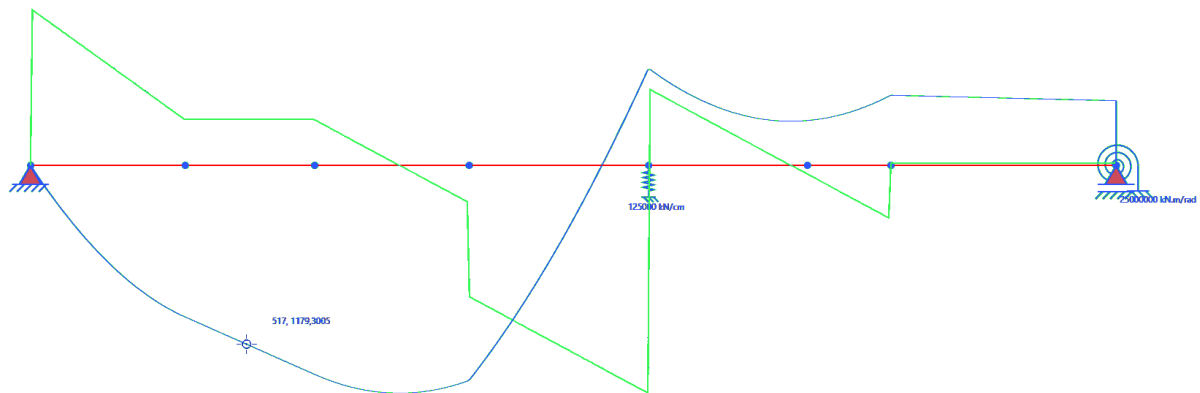
Uma vez produzidos os elementos de resultados da estrutura, o usuário pode então utilizar a área interativa para análise dos dados com uso do *mouse*. Um cursor adicional irá aparecer sobre as linhas graficadas dos esforços cortante e de flexão, exibindo ao lado do cursor os valores das solicitações e a posição na viga pela menor distância ao posicionamento do *mouse*. Ainda, na barra de controles à esquerda, é possível filtrar os gráficos exibidos para facilitar a análise dos resultados. A figura a seguir mostra como as solicitações de cortante e de flexão são exibidos no programa, bem como o cursor com informações dos dados sobre cada ponto.

Figura 22 - Exibição gráfica das condições inseridas para o cálculo de solicitações.



(fonte: elaborada pelo autor.)

Figura 23 – Exibição gráfica das solicitações calculadas pelo programa.



(fonte: elaborada pelo autor.)

9.2 ALGORITMOS DE CÁLCULO

O objetivo aqui não é ensinar a linguagem de programação utilizada, no entanto considera-se necessária uma breve explicação dos mecanismos utilizados pela linguagem para possibilitar um melhor entendimento dos algoritmos desenvolvidos e a forma que se relacionam.

9.2.1 Ambiente de Desenvolvimento

A plataforma utilizada para desenvolvimento do programa foi a .NET, que permite a criação de aplicativos e serviços para diversos dispositivos e sistemas operacionais com alta produtividade e desempenho. A ferramenta para compilação e uso das linguagens para desenvolvimento utilizada foi o Visual Studio 2015. Com essa combinação, é possível criar aplicativos com grande desempenho além de ser portátil para qualquer tipo de sistema. É possível programar dessa forma para sistemas Windows, iOS, Android, entre muitos outros dentro de uma mesma sistemática.

A Windows Presentation Foundation (WPF) por sua vez, é um subsistema gráfico da .NET que prove a esse ambiente de desenvolvimento, em linguagem XAML, recursos de interface de usuário dentro de ambientes Windows (para computadores ou celulares). Isso significa que todos os recursos de bibliotecas para execução das interfaces já existem no próprio sistema operacional sem necessidade da instalação de bibliotecas adicionais para execução do programa. Além disso, o WPF unifica diversos elementos de interface como renderização 2D e 3D, documentos, tipografia, gráficos vetoriais, animações em tempo de execução entre outros.

Todos esses recursos estão disponíveis gratuitamente no site da Microsoft, bem como sua documentação. Baseado nesse ambiente então é possível escrever os códigos em diversas linguagens, escolhido o C# para esse trabalho no que diz respeito aos algoritmos, e XAML para elementos de interface visto a escolha do WPF.

9.2.1.1 Organização da Linguagem C#

A linguagem C# utilizada é orientada a objetos, o que permite encapsulamento, herança e polimorfismo.

Encapsulamento é um recurso muito importante, pois este permite agrupar em um único objeto propriedades, métodos e outros membros daquele objeto. Por exemplo, é possível criar um objeto chamado “Concreto” e atribuir a esse métodos e propriedades. Isso significa que as propriedades podem ser sua resistência característica à compressão f_{ck} , tipo de agregado, idade, entre outros, e um método pode calcular e retornar resultados com base nessas propriedades

como módulo de elasticidade, deformação última, tensão para dada deformação, e assim por diante.

A herança significa a possibilidade de criar novas classes de objetos sobre aquelas já existentes. Por exemplo, uma classe já existente no ambiente para renderizar descrições geométricas em 3D pode ser ampliada para em um comando, fornecendo a seção da viga e seu comprimento, seja possível obter a visualização da viga tridimensionalmente.

Por fim, o polimorfismo significa que uma mesma classe ou método pode ser implementada de diferentes formas. Digamos que exista o método para calcular a resistência à tração do concreto. Este pode ser chamado sem indicação de nenhuma idade, retornando o valor de f_{ct} baseado em f_{ck} , ou caso informada uma idade, retornar o valor de f_{ctj} baseado em f_{cj} .

9.2.2 Implementação Da Seção Transversal da Viga

Para a seção da viga, conforme mencionado anteriormente, os parâmetros que definem sua geometria são D1 a D6, W1 a W4, T1, T2 e C1, para seção I, e B e H para seção retangular.

Criou-se uma classe para definição de objetos de seção transversal chama de *SectionGeometry*, com os parâmetros mencionados e os métodos *Area*, *YG*, *InertiaX*, *Ws*, *Wi* e *Coordinates*. Dessa forma, a partir apenas dos parâmetros geométricos é possível obter-se as propriedades de área, centro geométrico vertical, inércia em torno do eixo horizontal que passa pelo centro geométrico da peça, módulo de resistência a flexão superior e inferior, além de obter-se as coordenadas geométricas dos vértices da seção pelos métodos criados na mesma ordem em que estão mencionados aqui.

A rotina para cálculo das coordenadas verifica o tipo de parâmetros geométricos definidos no objeto, e então calcula todas as coordenadas da seção a partir deles, retornando uma lista de pontos em coordenadas x e y. A partir dessas coordenadas é possível calcular os outros parâmetros mencionados, bem como fornecer a outras rotinas, como a de representação gráfica da viga, informações para o seu desenho.

A implementação da classe *SectionGeometry* resumida está apresentada a seguir, com a definição de suas funções internas nas seções que seguem:

```
public class SectionGeometry
{
    // I Section Parameters
    public double D1;
    public double D2;
    public double D3;
    public double D4;
    public double D5;
    public double D6;
    public double D7;
    public double W1;
    public double W2;
    public double W3;
    public double W4;
    public double W5;
    public double T1;
    public double T2;
    public double C1;

    // Rectangular Section Parameters
    public double B;
    public double H;

    // Area Method
    public double Area()...

    // Centroid Method
    public double YG()...

    // Second Moment of Area Method
    public double InertiaX()...

    // Top Section Modulus Method
    public double Ws()...

    // Bottom Section Modulus Method
    public double Wi()...

    // Parameters to Coordinates Method
    public PointCollection Coordinates()...
}
```

9.2.2.1 Cálculo da Área

Foi implementada a fórmula de Shoelace, um caso particular do Teorema de Green, também conhecida como a fórmula de área de Gauss. Ela calcula a área de poligonais considerando apenas seus vértices como entrada. Essa implementação pode não funcionar para polígonos que possuem cruzamento de arestas ou sobreposição de áreas, caso que não ocorre no caso de seções de vigas e ainda, visto que as coordenadas são pré-calculadas pelo próprio programa, essa fórmula é absolutamente apropriada e muito eficiente para essa tarefa. A fórmula de Schoelace está apresentada na Equação (177) abaixo:

$$A = \frac{1}{2} \left| \sum_{i=1}^{n-1} x_i y_{i+1} + x_n y_1 - \sum_{i=1}^{n-1} x_{i+1} y_i - x_1 y_n \right| \quad (177)$$

Onde,

A: área do polígono formado pela união dos vértices na ordem em que estes aparecem;

x_i : valor da abscissa x na posição i da lista de coordenadas;

y_i : valor da ordenada y na posição i da lista de coordenadas;

n : número total de vértices do polígono.

Implementação do algoritmo:

```
public double Area()
{
    PointCollection _pList = Coordinates();
    double sumP = 0, sumN = 0;
    for (int i = 0; i < _pList.Count - 1; i++)
    {
        sumP = sumP + _pList[i].X * _pList[i + 1].Y;
        sumN = sumN - _pList[i + 1].X * _pList[i].Y;
    }
    return (double)(Math.Abs((sumP + sumN) / 2));
}
```

9.2.2.2 Cálculo do Centro Geométrico Vertical

Para determinação do centro geométrico vertical da seção transversal da viga aplicam-se as mesmas restrições existentes para o cálculo da área; isto é, não é permitido o cruzamento de arestas, e assim apenas pelas coordenadas dos vértices é possível obter-se o valor do centro. A fórmula do centroide para o polígono foi obtida do site da Wikipedia, descrita abaixo:

$$y_g = \frac{1}{6A} \sum_{i=0}^{n-1} (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad (178)$$

Onde,

y_g : centro geométrico vertical do polígono;

A: área do polígono formado pela união dos vértices na ordem em que estes aparecem;

x_i : valor da abscissa x na posição i da lista de coordenadas;

y_i : valor da ordenada y na posição i da lista de coordenadas;

n: número total de vértices do polígono.

O método que calcula o y_g da seção transversal da viga utiliza o método da área criado anteriormente, e sua implementação é conforme segue:

```
public double YG()
{
    double CY = 0;
    PointCollection _pList = Coordinates();
    for (int i = 0; i < _pList.Count - 1; i++)
    {
        CY = CY + (_pList[i].Y + _pList[i + 1].Y) * (_pList[i].X * _pList[i
+ 1].Y - _pList[i + 1].X * _pList[i].Y);
    }
    return (double)(1 / (Area() * 6) * CY);
}
```

9.2.2.3 Cálculo do Momento de Inércia de Área em Torno do Eixo Horizontal

O cálculo do momento de inércia de área em torno do eixo horizontal que passa pelo centroide da seção também é determinado com base apenas nas coordenadas da seção, geradas a partir de seus parâmetros geométricos. Assim, a fórmula para o cálculo adotada, segundo Brouke, Paul, é:

$$I_x = \frac{1}{12} \sum_{i=0}^1 (x_i^2 + x_i x_{i+1} + x_{i+1}^2) (x_i y_{i+1} - x_{i+1} y_i) \quad (179)$$

Onde,

I_x : momento de inércia de área em torno do eixo x passando na ordenada zero;

x_i : valor da abscissa x na posição i da lista de coordenadas;

y_i : valor da ordenada y na posição i da lista de coordenadas;

n: número total de vértices do polígono.

No entanto, essa fórmula depende da ordem em que os pontos são lidos, além de que seu resultado será em relação a ordenada zero do sistema de coordenadas, necessitando assim suplementarmente a aplicação do teorema de Steiner, conforme equação abaixo:

$$I_{xg} = I_x + A \cdot r^2 \quad (180)$$

Onde,

I_{xg} : momento de inércia de área em torno do eixo x que passa pelo centroide da área do polígono;

I_x : momento de inércia de área em torno do eixo x que passa na ordenada zero;

A: área da seção;

r: distância do eixo em que foi calculado I_x ao centroide da seção.

A implementação do algoritmo para cálculo do momento de inércia de área resultante ficou como:

```

public double InertiaX()
{
    Double ICGX = 0;
    PointCollection _pList = Coordinates();
    for (int i = 0; i < _pList.Count - 1; i++)
    {
        ICGX = ICGX + (Math.Pow(_pList[i].Y, 2) + _pList[i].Y * _pList[i + 1].Y + Math.Pow(_pList[i + 1].Y, 2)) * (_pList[i].X * _pList[i + 1].Y - _pList[i + 1].X * _pList[i].Y);
    }
    return (double)(ICGX / 12 - Area() * Math.Pow(YG(), 2));
}

```

9.2.2.4 Cálculo dos Módulos Resistentes à Flexão

Os módulos resistentes à flexão são calculados pela divisão do momento de inércia calculado em *InertiaX*, dividido pela distância do centro geométrico à extremidade superior ou inferior da seção da viga, conforme deseje-se W_s ou W_i respectivamente.

9.2.3 Implementação dos Carregamentos da Viga

Todos os carregamentos adicionados sobre a viga são armazenados em uma lista de elementos do tipo *LoadElement*, implementada com visibilidade global dentro do programa. A variável que armazena todas as cargas é chamada *loads*. *LoadElement* é uma classe que possui diferentes propriedades para armazenar todas as informações necessárias sobre cada carga individualmente.

Para possibilitar que cargas pontuais, uniformes ou ainda outros tipos sejam armazenados em uma única classe, definiu-se a propriedade *loadType*, atribuindo-se 0 para cargas pontuais (na direção y ou em torno de z), 1 para cargas uniformes (em y) e 2 para cargas lineares (em y).

Os parâmetros F_i e F_j podem assumir diferentes significados conforme o tipo da carga (*loadType*) e são interpretados em acordo em outras rotinas do programa que virem a utilizar os dados contidos nessa classe. Para o tipo 0 F_i armazena uma carga vertical nodal e F_j um

momento de flexão em torno de z . Caso algum desses valores seja zero, o programa considera que a carga é apenas de um tipo ou de outro, e caso contrário que existem os dois tipos. No caso de *loadType* valer 1, importa apenas o valor de *Fi* que será o valor da carga uniforme a ser distribuída na direção y .

Os parâmetros i e j dão indicação dos nós do *grid* a que essas cargas estão ligadas, sendo j utilizado apenas para cargas distribuídas uniformes ou lineares de modo que i indica início e j fim da distribuição. Para cargas nodais i corresponde ao próprio nó em que essas cargas estão aplicadas.

A propriedade *timeStamp* relaciona cada carga com o instante de tempo em que começa sua atuação ao longo da vida do concreto, de modo que cada carga começa no instante do *timeStamp* indicado em si, e termina no próximo instante de tempo existente na estrutura criada para esse fim.

Outras propriedades nessa classe não possuem função de cálculo, sendo apenas para representação visual dos elementos e para armazenamento de seus modelos tridimensionais.

Segue a implementação da classe:

```
public class LoadElement
{
    public int loadType { get; set; }
    public double Fi { get; set; }
    public double Fj { get; set; }
    public int i { get; set; }
    public int j { get; set; }
    public int timeStamp { get; set; }
    public Model3DGroup model = new Model3DGroup();
    public List<ModelVisual3D> label = new List<ModelVisual3D>();
    public bool onScreen = false;
}
```

9.2.4 Implementação das Restrições Nodais

Todos os apoios inseridos na viga são armazenados na variável global chamada *supportCollection*. Essa variável é do tipo *SupportElement*, criada com as propriedades

necessárias para identificação do tipo de apoio, mola, ou cedimento, bem como seu modelo para visualização.

A implementação da classe é:

```
public class SupportElement
{
    public double springX { get; set; }
    public double springY { get; set; }
    public double springZ { get; set; }
    public double dX { get; set; }
    public double dY { get; set; }
    public double thetaZ { get; set; }
    public bool fixedX { set; get; }
    public bool fixedY { set; get; }
    public bool fixedZ { set; get; }
    public int gridNumber { get; set; }
    public Model3DGroup model = new Model3DGroup();
}
```

Os parâmetros *springX*, *springY* ou *springZ* armazenam o valor das rigidezes das molas em cada uma das direções identificadas pela última letra do nome da propriedade. Os parâmetros *dX*, *dY* e *thetaZ* indicam deslocamentos pré-definidos em cada uma das direções x, y ou z respectivamente. As propriedades *fixedX*, *fixedY* ou *fixedZ* registram a existência de vínculos ideais em cada uma das direções. Por fim, a variável *gridNumber* informa o número do *grid* em que o apoio se localiza.

9.2.5 Implementação dos Nós

Para os nós foram criados os elementos chamados de *grid* que possuem duas variáveis distintas no código do programa. Uma lista do tipo *double* armazena a posição do *grid* ao longo da viga, e uma outra do tipo *GridModel*, criada apenas para armazenar dados do modelo visual desses grids.

9.2.6 Implementação dos Materiais

O uso de linguagem orientada a objeto permite uma facilidade de implementação das diferentes propriedades da norma NBR 6118:2014 conforme cada material, e o polimorfismo do código ainda permite que mesmo uma equação que tenha diferentes formas de cálculo conforme a situação possa receber parâmetros diversos fornecendo respostas adequadas.

9.2.6.1 Concreto

O concreto dentro do programa foi armazenado na variável *concreto* definida pela classe *Concrete* criada para tal, cuja implementação abreviada está apresentada a seguir:

```

public partial class Concrete
{
    // Concrete Properties
    public string name { get; set; }
    public double fckMPA { get; set; }
    public double alfaE { get; set; }
    public int concreteType { get; set; }
    public double gammaC { get; set; }
    public int CAA { get; set; }
    public double humidityAir { get; set; }

    // Young Modulus Method
    public double Eci()...

    // Young Modulus ate the age of j days Method
    public double Eci(int _age)...

    // Concrete compression strength ate j days of age Method
    public double fc_t(int _age)...

    // Concrete linear-parabolic deformation limit Method
    public double epsilonC2()...

    // Concrete ultimate deformation limit Method
    public double epsilonCU()...

    // Concrete compression stress for given deformation at the age of j days
Method
    public double sigmaC(double _ec, int _age)...

    // Secant Concrete Modulus at j days of age Method
    public double Ecs(int _age)...

    // λ parameter Method
    public double lambda()...
}

```

Nessa classe existem propriedades fornecidas pelo usuário do programa, que são gravadas então em *name*, *fckMPA*, *alfaE*, *concreteType*, *gammaC*, *CAA* e *humidityAir*. Cada uma dessas propriedades é definida em acordo com a NBR 6118:2014. O usuário define essas propriedades selecionando valores de listas pré-definidas no código, a exceção de *humidityAir* em que o valor é lido de um campo digitado pelo engenheiro, e que representa a umidade relativa do ar em que o concreto em questão estará exposto, e de *fckMPA* que é deduzida a partir da variável *name*.

Para as demais propriedades, as listas são conforme segue:

```
List<string> cList = new List<string> { "C20", "C25", "C30", "C35", "C40",
"C45", "C50", "C55", "C60", "C65", "C70", "C75", "C80", "C85", "C90" };

List<string> cAAList = new List<string> { "CAA-I (Weak)", "CAA-II
(Moderated)", "CAA-III (Strong)", "CAA-IV (Very Strong)" };

List<string> agregateList = new List<string> { "Basalt or Diabase", "Granite
or Gneiss", "Limestone", "Arenite" };

List<string> CTypeList = new List<string> { "CP-I", "CP-II", "CP-III", "CP-
IV", "CP-V-ARI" };

List<string> gammaCList = new List<string> { "Normal (γc = 1.4)", "Special
or Construction (γc = 1.2)", "Exceptional (γc = 1.2)" };
```

Sendo, *cList* a classe do concreto que dará valor à propriedade *name*, e também ao *fckMPA* extraíndo a primeira letra “C” do valor e considerando o número que segue como o valor para o f_{ck} em MPa para aquele concreto.

A *cAAList* define a classe de agressividade ambiental no parâmetro CAA, traduzindo os valores da lista para números de 1 a 4 conforme a classe; *agregateList* irá atribuir valores ao *alfaE* de 1,2, 1,0, 0,9 ou 0,7 conforme norma para α_e ; a *CTypeList* registrará valores de 1 a 5 para cada tipo de Cimento Portland escolhido, na variável *concreteType*, que por sua vez definirá o parâmetro s nos métodos internos da classe com base nessa propriedade, conforme norma.

Método $f_{ct}(age)$

Esse método retorna o valor de f_{ckj} , calculado conforme descrito no capítulo 8.

Métodos Eci() e Eci(age)

Esses métodos retornam o módulo de elasticidade do concreto em função de uma idade ou puramente, conforme equações prescritas em norma.

Métodos epsilonC2() e epsilonCU()

Esses métodos retornam os valores de ϵ_{c2} e ϵ_{cu} para o concreto com base nas suas propriedades, de acordo com a NBR 6118:2014, para concretos do grupo I (f_{ck} até 50 MPa) e grupo II (f_{ck} de 55 a 90 MPa).

Método sigmaC(ec, age)

Esse método calcula a tensão de compressão resistente do concreto, σ_c , ainda não reduzida por nenhum coeficiente, com base em uma deformação $_{ec}$, correspondente a ϵ_c e idade $_{age}$, correspondente a t .

Método Ecs(age)

Retorna o valor do módulo de elasticidade secando do concreto para determinada idade, informada em $_{age}$.

Método $\lambda()$

Retorna o valor λ conforme norma para o concreto definido pelo usuário.

9.2.7 Implementação do Método da Rigidez

Para a implementação do método da rigidez foram criadas uma série de classes para armazenamento das estruturas, bem como rotinas para sua solução.

Primeiramente fez-se a composição de um sistema matricial composto por três elementos: matriz de rigidez, vetor de deslocamentos e vetor de forças. Cada um desses elementos por sua vez, possui suas próprias rotinas e métodos.

Após, foram elaboradas rotinas para a solução desse sistema.

9.2.7.1 Implementação do Sistema Matricial

O sistema matricial definido é composto pela matriz de rigidez, vetor de deslocamentos e vetor de forças, conforme descrito no capítulo 3, equação (1).

Matriz de Rigidez

Foi implementada a classe KMatrix com propriedade *elements*, que armazena cada uma das posições da matriz k_{ij} em uma lista bidimensional e um método chamado *Refresh*, que realiza o preenchimento dos elementos da matriz com base em um vetor de comprimentos chamado L, o módulo de elasticidade para a estrutura E, a área da seção transversal A, e a inércia de área em torno do eixo de flexão chamado de I. Além dessa propriedade e método, a estrutura possui outras variáveis de uso interno para processamento dos dados.

Dessa forma, é possível obter-se a matriz de rigidez da estrutura por um único método, fornecendo os parâmetros facilmente obtidos pelas demais classes criadas como E, chamando *Eci()* da classe Concrete, A pelo comando *Area()* e I pelo comando *InertiaX()*, ambos da classe *SectionElement*. Para a lista L, uma rotina em separado relaciona os dados dos *grids* (*gridList*) com os dados dos apoios (*supportCollection*) determinando o comprimento dos trechos entre apoios.

A rotina interna do método *Refresh*, sendo n o número de elementos em L , irá gerar uma matriz de $[3*(n+1)] \times [3*(n+1)]$.

```
public class KMatrix // Matrix containing the stiffness for each node of the beam.
{
    public List<List<double>> elements = new List<List<double>>();
    int resto1, resto2;
    double partial;
    int iL, jL;
    public void Refresh(List<double> L, Double E, Double A, Double I)...
}
```

Vetor de Deslocamentos

Foi criada a classe *dVector* com três propriedades, *elementSpring*, *elementsFixed*, *elementsDisplacement* e um método *Refresh*. O método *Refresh* recebe a lista com o comprimento dos trechos L , e a lista dos apoios de *supportCollection*. Com base nesses dois elementos esse método irá preencher as demais propriedades armazenando no vetor *elementSpring* as rigidezes das molas inseridas pelo usuário, em *elementsFixed* as informações de nós restringidos por apoios ideais, e em *elementsDisplacement* as informações de deslocamentos pré-determinados pelo usuário.

Posteriormente, para solução do sistema a matriz de rigidez e o vetor de forças sofrem simplificações, e cada uma das propriedades de *dVector* irá ser processada de uma maneira diferente, conforme descrito no capítulo 3. A propriedade *elementSpring* irá somar as rigidezes contidas nela em posições específicas da matriz de rigidez, anulando o valor da mesma linha correspondente no vetor de forças; *elementsFixed* irá zerar a posição da linha no vetor de forças; *elementsDisplacement* provocarão a subtração das rigidezes de cada linha no vetor de forças multiplicadas pelo valor do próprio deslocamento, das rigidezes em coluna de mesmo número que a linha em que aparece tal deslocamento.

```
public class dVector // The displacement vector containing dx, dy and thetaz.
{
    public List<double> elementsSpring = new List<double>();
    public List<bool> elementsFixed = new List<bool>();
    public List<double> elementsDisplacement = new List<double>();

    public void Refresh(List<int> _nosMatriz, List<SupportElement>
    _supportList)...
}
```


Vetor de Forças

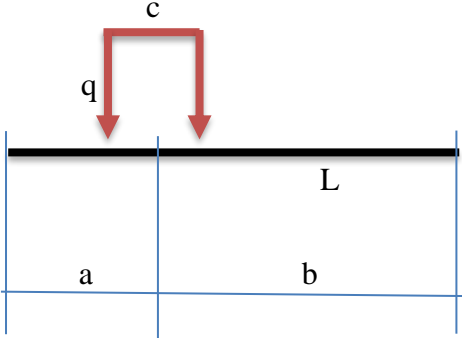

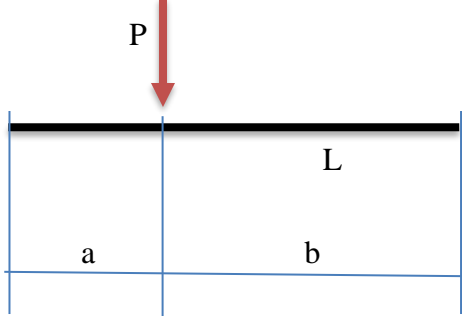
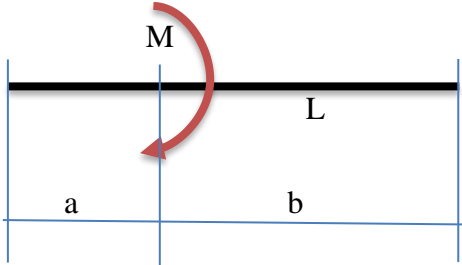
Para o vetor de forças foi implementada a classe *fVector*, composta pela propriedade *elements* e pelo método *Refresh*. Em *elements* serão armazenados os valores das forças de cada nó, em sua direção correspondente a cada linha, sendo que o método *Refresh* provocará a determinação desses valores e seu armazenamento em *elements*.

Para determinação desse vetor o método *Refresh* recebe a lista de comprimentos dos trechos *_nosMatriz*, a lista de cargas aplicadas sobre a estrutura *_loads*, o instante de tempo em que se deseja considerar as cargas *_timePoint*, e a lista de *grids* *_gridPoints*, necessária pois as cargas não se localizam necessariamente sobre os apoios, podem estar no meio de um vão da viga.

```
public class fVector // The forces vector containing Vx, Vy and Mz.
{
    public List<double> elements = new List<double>();
    public void Refresh(List<int> _nosMatriz, List<LoadElement> _loads, int
_timePoint, List<double> _gridPoints)...
}
```

O algoritmo desenvolvido para determinação desse vetor em *Refresh* aplica o método dos deslocamentos, de modo a transformar todas as cargas que estão entre os vãos em cargas nodais verticais e de momento. Para esse trabalho foram previstas cargas nodais e distribuídas constantes. Em geral tabelas com essas equações preveem particularizações de equações mais gerais, não consideradas aqui visto que uma carga distribuída, por exemplo, que vá de um nó ao outro do trecho terá resultado idêntico ao da equação particularizada, de modo que as equações aplicadas foram mais genéricas, conforme apresentadas na Tabela 7.

Tabela 7 - Equivalência das cargas nodais pelo método dos deslocamentos.

	 $M_A = \frac{qc}{12L^2} [12ab^2 + c^2(L - 3b)]$ $M_B = -\frac{qc}{12L^2} [12a^2b + c^2(L - 3a)]$ $V_A = \frac{qc}{12L^3} [12ab(b - a) + 3c^2(a - b) + 12bL^2]$ $V_B = \frac{qc}{12L^3} [12ab(a - b) + 3c^2(b - a) + 12aL^2]$
	$M_A = \frac{Pab^2}{L^2}$ $M_B = -\frac{Pa^2b}{L^2}$ $V_A = \frac{P(b^3 + 3ab^2)}{L^3}$ $V_B = \frac{P(a + 3a^2b)}{L^3}$
	$M_A = -M \frac{b}{L} (2 - 3b)$ $M_B = -M \frac{a}{L} (2 - 3a)$ $V_A = \frac{-M + M_A + M_B}{L}$ $V_B = -\frac{-M + M_A + M_B}{L}$

(fonte: elaborada pelo autor.)

Condições de Contorno

Uma rotina chamada *mountKMatrix* foi desenvolvida para computar todo o sistema (matriz de rigidez, vetor de deslocamentos e vetor de forças) foi desenvolvida, de modo que chamando *mouuntKMatrix* com as variáveis “concreto”, “gridPoints”, “supportCollection”, “section”, “loads” e instante de tempo selecionado, todo o sistema é retornado em uma estrutura única do tipo *MatrixSystem*, de implementação abaixo:

```
public class MatrixSystem
{
    public KMatrix matrixK = new KMatrix();
    public dVector vectorD = new dVector();
    public fVector vectorF = new fVector();
}
```

As condições de contorno são então aplicadas sobre esse sistema utilizando os elementos do tipo *dVector.elementsSpring* com valor de rigidez das molas inseridas em apoios (corrigidas suas unidades de metros para centímetros onde houver), que são somadas com os elementos de mesma linha e coluna na variável *matrixK*. Ainda, para cada elemento do tipo *dVector.elementsDisplacement*, que contém informações de vínculos ideais e cedimentos, cada rigidez na coluna correspondente a linha do vínculo ideal é multiplicada pelo valor do deslocamento definido e subtraída no vetor de forças, além de serem zeradas as posições nessa mesma coluna que não a da linha do elemento em questão, sendo a esta atribuída então o valor 1 (equivalente a eliminar a equação correspondente a essa linha do sistema). A implementação das condições de contorno é conforme segue, sendo a variável do tipo *MatrixSystem* chamada de *matrixSystem*:

```
for (int i = 0; i < matrixSystem.vectorD.elementsSpring.Count; i++)
{
    if ((matrixSystem.vectorD.elementsSpring[i] != 0) &&
(matrixSystem.vectorD.elementsFixed[i] == false))
    {
        int resto;
        Math.DivRem((i + 1), 3, out resto);
        if(resto == 0)
            matrixSystem.matrixK.elements[i][i] =
matrixSystem.matrixK.elements[i][i] + matrixSystem.vectorD.elementsSpring[i] * 1000;
        else
            matrixSystem.matrixK.elements[i][i] =
matrixSystem.matrixK.elements[i][i] + matrixSystem.vectorD.elementsSpring[i] / 100;
```

```

    }
    if ((matrixSystem.vectorD.elementsFixed[i] == true))
    {
        for (int j = 0; j < matrixSystem.matrixK.elements[i].Count;
j++)
        {
            if (i != j)
            {
                matrixSystem.vectorF.elements[j] =
matrixSystem.vectorF.elements[j] - matrixSystem.matrixK.elements[i][j] *
matrixSystem.vectorD.elementsDisplacement[i];
                matrixSystem.matrixK.elements[i][j] = 0;
                matrixSystem.matrixK.elements[j][i] = 0;
            }
            else
                matrixSystem.matrixK.elements[i][j] = 1;
        }
    }
}

```

Obtenção dos Deslocamentos Nodais

Após a montagem do sistema matricial e aplicadas as condições de contorno, o programa resolve o sistema matricial do tipo $[A] \cdot \{x\} = \{B\}$, encontrando o valor das incógnitas em x (vetor de deslocamentos). Para solução das equações implementou-se um algoritmo de eliminação Gaussiana do sistema, conforme segue:

```

private List<double> solveMatrix(List<List<double>> A, List<double> B)
{
    List<double> X = new List<double>();
    for (int k = 0; k < A.Count; k++)
    {
        for (int i = k + 1; i < A.Count; i++)
        {
            double f = A[i][k] / A[k][k];
            for (int j = k + 1; j < A.Count; j++)
            {
                A[i][j] = A[i][j] - A[k][j] * f;
            }
            B[i] = B[i] - B[k] * f;
            A[i][k] = 0;
        }
    }
    for (int i = 0; i < A.Count; i++)
        X.Add(0);
    for (int i = X.Count - 1; i >= 0; i--)
    {
        double soma = 0;
        for (int j = i + 1; j < A.Count; j++)
        {
            soma = soma + A[i][j] * X[j];
        }
    }
}

```

```

    }
    X[i] = (B[i] - soma) / A[i][i];
  }
  return X;
}

```

Após obter-se o vetor x da equação, é realizada uma substituição dos deslocamentos que estavam previamente definidos pelas informações dos apoios inseridas pelo usuário, gerando-se assim o vetor com os deslocamentos finais dos nós. Os deslocamentos finais ficam armazenados na variável *deslocs* do tipo *dVector*, em que os valores estão na propriedade *elementDisplacement*.

Obtenção das Reações nos Apoios

Uma vez obtidos os deslocamentos finais dos nós, e com a matriz de rigidez inicial (antes da aplicação das condições de contorno), as forças de reação podem ser obtidas pela multiplicação da matriz de rigidez pelo vetor de deslocamentos. As reações são então armazenadas na variável *reactions*, do tipo *List<double>*. A própria ordem da lista indica sua posição, seguindo a mesma ordem dos apoios.

9.2.8 Implementação das Solicitações

As solicitações são calculadas a partir das reações obtidas em *reactions*, das informações de carregamentos em *loads*, das informações de apoios em *supportCollection*, dos *grids*, e de um dado “x” fornecido. Inicialmente um visualizador gráfico foi implementado que desenha em duas dimensões os valores para as solicitações percorrendo x do início ao fim da viga. Após, os valores de x serão lidos a partir do apontamento com o mouse pelo usuário sobre esse gráfico gerado, que irá então chamar as funções de cálculo das solicitações para retornar valores com precisão.

Uma rotina determina a ordem entre apoios e carregamentos, bem como traduz índices que indicavam números referentes a *grids* em coordenadas de referência, chamadas de *xValue* dentro de cada nova estrutura criada para armazenar esses dados. Após então as rotinas *calcQ* e *calcM*

calculam as solicitações de cortante e de momento de flexão, respectivamente. Segue o algoritmo implementado para cálculo das solicitações de cortante:

```

public double calcQ(double _x)
{
    double Q = 0;
    double Qp = 0;
    double Qr = 0;
    double Qt = 0;
    double Qd = 0;

    for (int i = 1 ; i < reactions.Count; i += 3)
    {
        if (_x >= supports[(i - 1) / 3].xValue)
        {
            Qr = Qr + reactions[i];
        }
    }

    for (int i = 0; i < loads.Count; i++)
    {
        if (loads[i].loadType == 0)
            if (_x >= grids[grids.FindIndex(x => x.gridNumber ==
loads[i].i)].xValue)
                Qp = Qp + loads[i].Fi;
        if (loads[i].loadType == 1)
        {
            double menor, maior, estepe;
            menor = grids[grids.FindIndex(x => x.gridNumber ==
loads[i].i)].xValue;
            maior = grids[grids.FindIndex(x => x.gridNumber ==
loads[i].j)].xValue;
            if(menor > maior)
            {
                estepe = menor;
                menor = maior;
                maior = estepe;
            }

            if ((_x >= menor) && (_x <= maior))
                Qd = Qd + loads[i].Fi * (_x - menor)/100;
            else if ((_x >= menor) && (_x >= maior))
                Qd = Qd + loads[i].Fi * (maior - menor)/100;
        }
    }
    Q = Qp + Qr + Qd + Qt;
    return Q;
}

```

A variável Q_r acumula as solicitações provenientes das reações, Q_p das cargas pontuais, Q_d de cargas distribuídas. Q_t destina-se a implementação futura de cargas triangulares, por exemplo.

Para solicitações de momento de flexão, segue implementação:

```

public double calcM(double _x)
{
    double M= 0;
    double Mp = 0;
    double Mr = 0;
    double Mt = 0;
    double Md = 0;

    for (int i = 1; i < reactions.Count; i += 3)
        if (_x >= supports[(i - 1) / 3].xValue)
            Mr = Mr + reactions[i] * (_x - supports[(i - 1) / 3].xValue);
    for (int i = 2; i <= reactions.Count; i += 3)
        if (_x >= supports[(i - 2) / 3].xValue)
            Mr = Mr - reactions[i];

    for (int i = 0; i < loads.Count; i++)
    {
        if (loads[i].loadType == 0)
            if (_x >= grids[grids.FindIndex(x => x.gridNumber ==
loads[i].i)].xValue)
                Mp = Mp + loads[i].Fi * (_x - grids[grids.FindIndex(x =>
x.gridNumber == loads[i].i)].xValue);
                Mp = Mp + loads[i].Fj;
            if(loads[i].loadType==1)
            {
                double menor, maior, estepe;
                menor = grids[grids.FindIndex(x => x.gridNumber ==
loads[i].i)].xValue;
                maior = grids[grids.FindIndex(x => x.gridNumber ==
loads[i].j)].xValue;
                if (menor > maior)
                {
                    estepe = menor;
                    menor = maior;
                    maior = estepe;
                }
                if ((_x >= menor) && (_x <= maior))
                    Mp = Mp + loads[i].Fi / 100 * Math.Pow((_x - menor), 2) / 2;
                else if ((_x >= menor) && (_x >= maior))
                    Mp = Mp + (loads[i].Fi / 100 * (maior - menor) * (_x - maior
+ (maior - menor) / 2));
            }
        }
        M = Mp + Mr + Md + Mt;
    }
    return M/100;
}

```

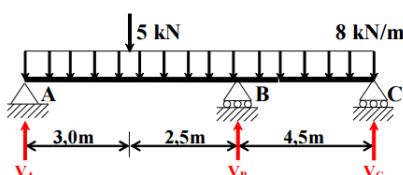
9.3 VALIDAÇÃO DOS RESULTADOS

Para validação dos resultados foi utilizado o programa ftool, no qual inseriram-se estruturas com parâmetros semelhantes aos do programa, com pequenas diferenças de precisão visto o ftool aceitar um número limitado de casas após a vírgula, ao passo que no programa os valores internos são calculados com precisão do tipo *double*, isto é, até dez dígitos de precisão e valores entre $-1,7 \times 10^{308}$ a $+1,7 \times 10^{308}$. Inserindo-se valores no ftool de diferentes ordens, como por exemplo, para momento de inércia, é possível inferir-se que sua precisão é do tipo *float*, isto é, números de $-3,4 \times 10^{38}$ até $+3,4 \times 10^{38}$, o que pode gerar pequenas diferenças nos resultados, no entanto, negligenciáveis.

Exemplo 1

O exemplo 1 foi tirado de uma lista de exercícios do site de Internet www.profwillian.com, com resolução analítica pelo método das forças. Então inseriram-se as informações do problema no ftool e no programa desenvolvido para este trabalho, referido aqui como PSB, para comparação dos resultados.

Figura 24 – Representação do problema resolvido no exemplo 1.



(fonte: lista de exercícios do prof. William.)

Tabela 8 – Comparação de resultados para o exemplo 1.

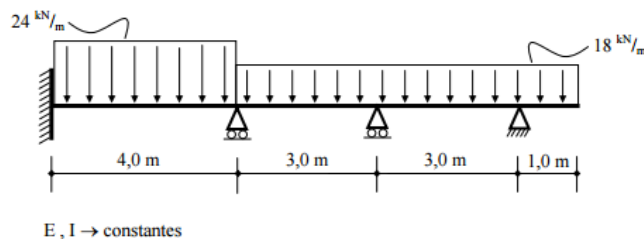
Nó	ftool		PSB	diferença	ftool		PSB	diferença
	Ry	kN			dy	mm		
A	19,064050	19,0640495868	0,0000%	0	0	-9,452025E-06	-9,45182221282721E-06	0,0021%
B	54,302112	54,3021120294	0,0000%	0	0	3,109610E-06	3,10954267739291E-06	0,0022%
C	11,633838	11,6338383838	0,0000%	0	0	2,194400E-06	2,19435251185440E-06	0,0022%

(fonte: elaborada pelo autor.)

Exemplo 2

Esse exemplo foi retirado de uma lista de exercícios em www.ims.eng.br.

Figura 25 – Representação do problema resolvido no exemplo 2.



(fonte: lista de exercícios do prof. Iberê.)

Tabela 9 – Comparação de resultados para o exemplo 1.

Nó	ftool PSB		diferença	ftool PSB	
	Ry kN	kN		Mz kN.m	kN.m
1	51,350000	51,3500000000	0,0000%	36,466667	36,466666666667
2	75,261111	75,2611111111	0,0000%	dy (mm)	
3	51,466667	51,4666666667	0,0000%	mm	mm
4	43,922222	43,9222222222	0,0000%	7,117831E-04	0
Nó	ftool PSB		diferença		
	θ_z rad	rad			
1	2,205293E-06	2,20524596312237E-06	0,0021%		
2	-8,681286E-07	-8,68109884736605E-07	0,0022%		
3	1,267221E-06	1,26719357582405E-06	0,0022%		
4					

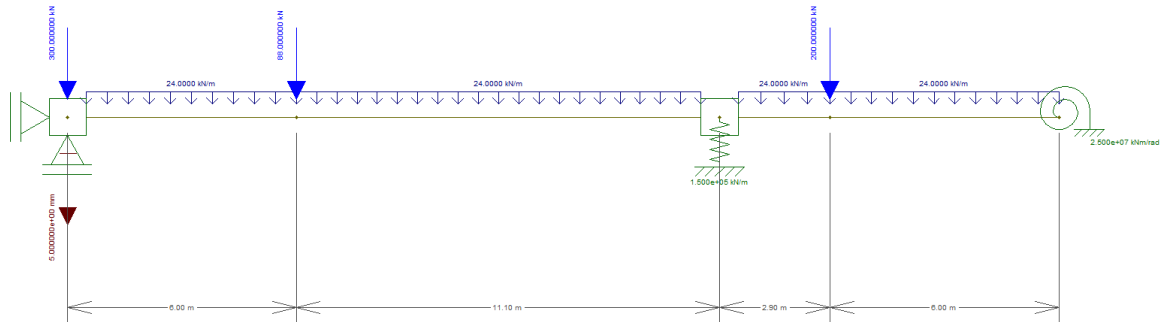
(fonte: elaborada pelo autor.)

Exemplo 3

Exemplo elaborado pelo autor para demonstrar uma viga com uso de molas e deslocamento pré-determinado em apoio. Buscou-se incluir nesse exemplo elementos que explorassem os diferentes tipos de condições de contorno que deveriam ser aplicadas no método da rigidez. Dessa forma, usou-se apoios com molas, apoio com deslocamento pré-definido, cargas pontuais

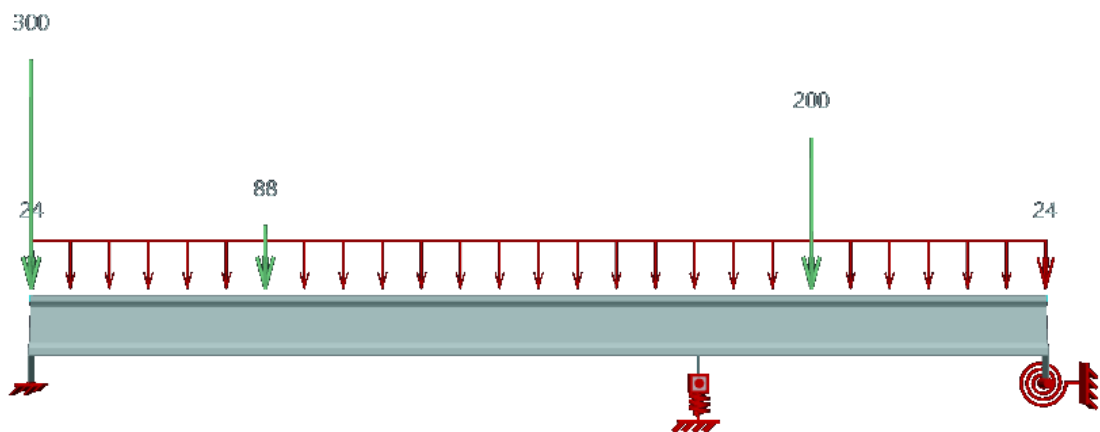
e distribuídas, além de apoios fixos. Todos os resultados mostraram-se compatíveis com os obtidos pelo ftool para essa mesma estrutura.

Figura 26 – Representação do exemplo 3 no ftool.



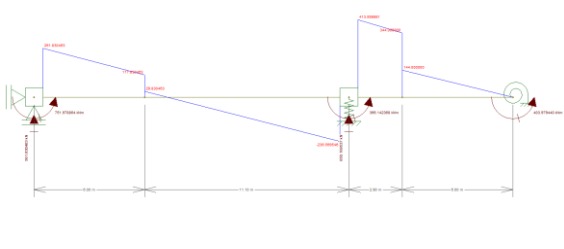
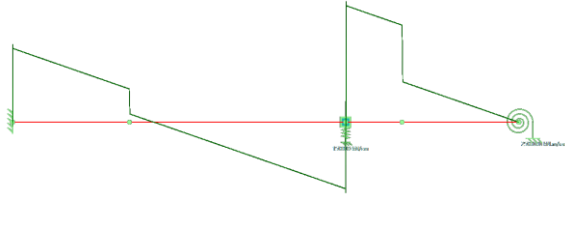
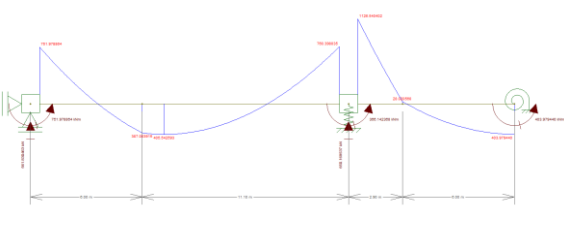
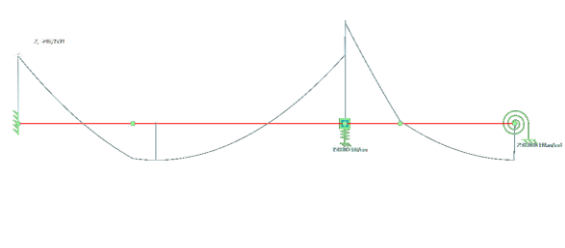
(fonte: elaborada pelo autor.)

Figura 27 – Representação do exemplo 3 no PSB.



(fonte: elaborada pelo autor.)

Tabela 10 – Comparação dos diagramas obtidos para as solicitações da viga.

	
Diagrama de esforço cortante no ftool.	Diagrama de esforço cortante no PSB.
	
Diagrama de momento de flexão no ftool.	Diagrama de momento de flexão no PSB.

(fonte: elaborada pelo autor.)

Tabela 11 – Comparação de resultados para o exemplo 3.

Nó	ftool		PSB	diferença	ftool		PSB	diferença
	Ry	kN			Mz	kN.m		
1	561,830463	561,8303268497	0,0000%	751,978864	751,977715617699	0,0000%		
2	650,169537	650,1696731503	0,0000%	366,142368	366,141541582634	0,0000%		
3	-	-	-	403,979440	403,979331929803	0,0007%		
Nó	ftool		PSB	diferença	ftool		PSB	diferença
	dy	mm			θz	rad		
1	-5,000000E+00	-5,000000000000000E+00	0,0000%	0,000000E+00	0,000000000000000E+00	0,0000%		
2	-4,334464E+00	-4,33446448766856E+00	0,0000%	0,000000E+00	0,000000000000000E+00	0,0000%		
3	-6,678410E+00	-6,67836221133987E+00	0,0007%	-1,615918E-05	-1,61591732771921E-05	0,0000%		

(fonte: elaborada pelo autor.)

10 CONSIDERAÇÕES FINAIS

O trabalho desenvolvido possibilitou uma profunda revisão da complexidade e diversidade dos assuntos envolvidos no desenvolvimento de um programa de computador para realização de projeto de vigas protendidas por pós-tração. Diferentes aspectos de disciplinas do curso de engenharia civil e além foram estudados e desenvolvidos para possibilitar a criação do conceito do *software* proposto. Partiu-se dos conceitos básicos da mecânica estrutural e de estudos das normas brasileiras, seguindo-se por um aprofundamento em áreas específicas entre a engenharia civil e a computação para obter-se um desenvolvimento de modelos sofisticados dos objetos em questão com parte de sua implementação em uma ferramenta que permitirá a propagação desses conhecimentos e o aprimoramento em futuros trabalhos acadêmicos. Os resultados obtidos mostraram-se válidos e servirão de poderosa ferramenta, não apenas na área aqui proposta, para a solução de vigas protendidas, mas de tantas outras áreas nas quais o ponto a que se chegou serve de partida para outros fins. A combinação da tecnologia de maneira sofisticada com interface moderna e interativa permitirá que sejam criados novos *softwares* cada vez mais avançados aumentando o desempenho e facilitando o trabalho dos engenheiros. Buscou-se demonstrar junto a esse trabalho como é possível fazer um programa com características de ferramentas comerciais mundialmente difundidas, em um trabalho acadêmico.

O programa desse trabalho servirá a fins acadêmicos por hora, pois é um trabalho recém começado há cerca de um ano atrás, um curto espaço de tempo, desenvolvido por apenas um indivíduo. Ferramentas comerciais são fruto do trabalho de grandes equipes, e ainda assim propensas a *bugs* e erros, sofrendo constante aprimoramento e o acompanhamento de sua empresa criadora, até que se obtenha um nível de experiência ao usuário satisfatório.

REFERÊNCIAS

GERE, J. M.; WEAVER JR., W. **Análise de Estruturas Reticuladas**, Rio de Janeiro: Editora Guanabara Dois S.A., 1981.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6.118**: Projeto de estruturas de concreto – Procedimento. Rio de Janeiro, 2014.

VERÍSSIMO, G. S.; CÉSAR JR., K. M. L. **Concreto Protendido: Perdas de Protensão**. Rio de Janeiro, 1998.

AALAMI, B. O. **Load Balancing: A Comprehensive Solution to Post-Tensioning**. Technical Paper – ACI Structural Journal, Title no. 87-S68, nov./dez. 1990.

MICROSOFT, DEVELOPER NETWORK – Windows Presentation Foundation Development Guide. Disponível em: <[https://msdn.microsoft.com/en-us/library/ms754130\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/ms754130(v=vs.110).aspx)>. Acesso em 19 de setembro de 2016.

SCAVASSIN, R. M. **Programa para Cálculo e Detalhamento de Armadura de Vigas Pré-Tracionadas**. 2012. 198 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Civil) - Universidade Federal São Carlos, São Carlos.

ART OF PROBLEM SOLVING – Shoelace Theorem. Disponível em: <http://www.artofproblemsolving.com/wiki/index.php?title=Shoelace_Theorem>. Acesso em 03 de novembro de 2016.

BOURKE, PAUL. Polygons & Meshes. Disponível em: <<http://paulbourke.net/geometry/polygonmesh/>>. Acesso em 03 de novembro de 2016.

WIKIPEDIA – Parallel axis theorem. Disponível em: <https://en.wikipedia.org/wiki/Parallel_axis_theorem>. Acesso em 03 de novembro de 2016.

WIKIPEDIA – Gaussian elimination. Disponível em: <https://en.wikipedia.org/wiki/Gaussian_elimination>. Acesso em 03 de novembro de 2016.

ROSA, W. A – Lista 02 de Exercícios de Hiperestática – Método das Forças – Gabarito. Disponível em: <http://www.profwillian.com/estruturas/Lista02-Hiperestatica-Metodo_das_Forcas_gab.pdf>. Acesso em 03 de novembro de 2016.

PROF. IBERÊ – Estática das Estruturas I – Método dos Esforços. Disponível em: <<http://www.ims.eng.br/upload/img/noticia/24042012154841968485893.pdf>>. Acesso em 03 de novembro de 2016.

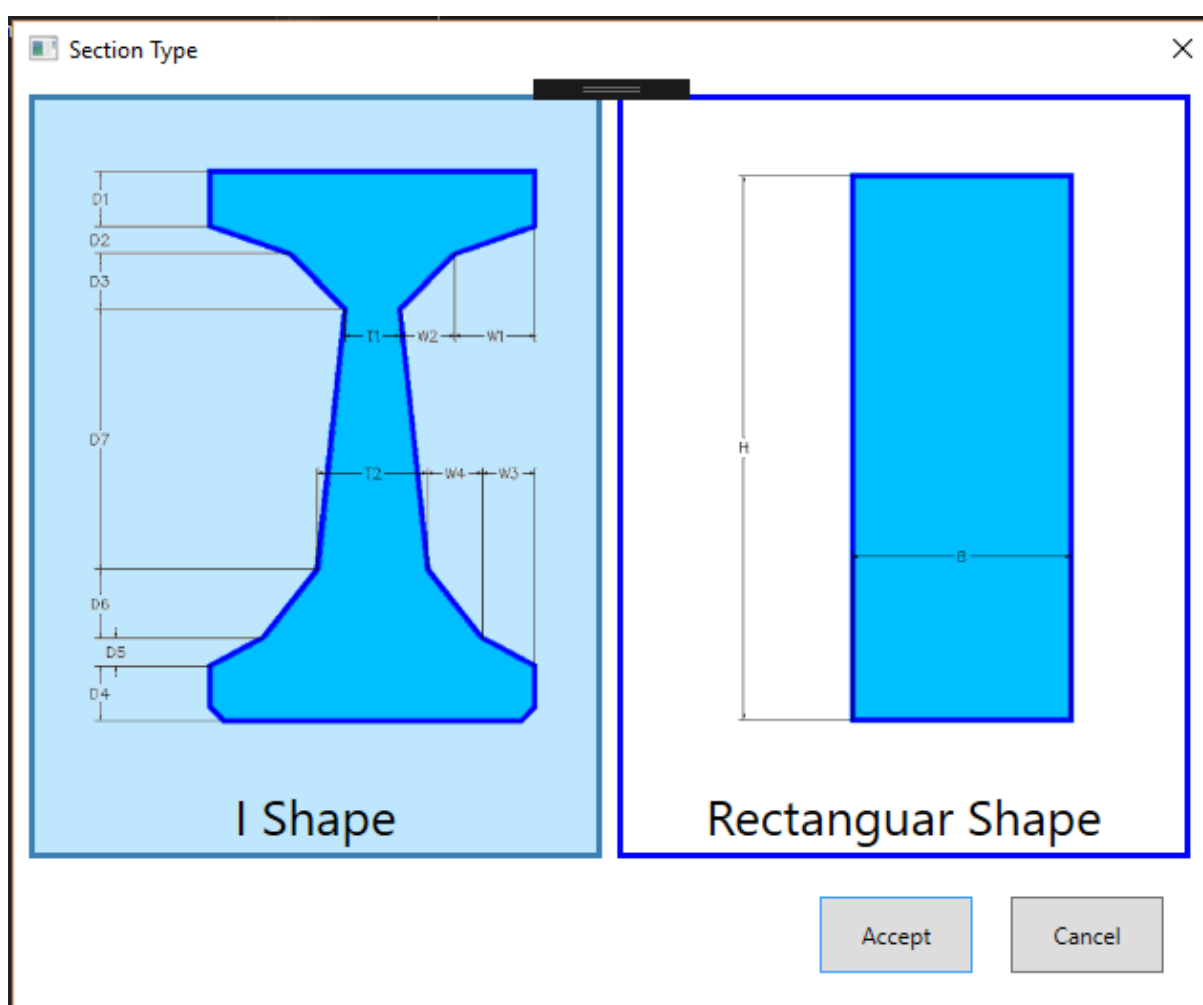
GAVIN, P. H. Frame Element Stiffness Matrices. CEE 421L. Matrix Structural Analysis. Fall 2014. Duke University, Duke, EUA.

ANEXO A – Manual do Programa

A.1 ESCOLHA DA SEÇÃO

Para escolher uma seção deve-se clicar sobre o botão *section* no grupo *Beam*, e então marcar na janela que abre a forma da seção do tipo I ou retangular, conforme figura abaixo:

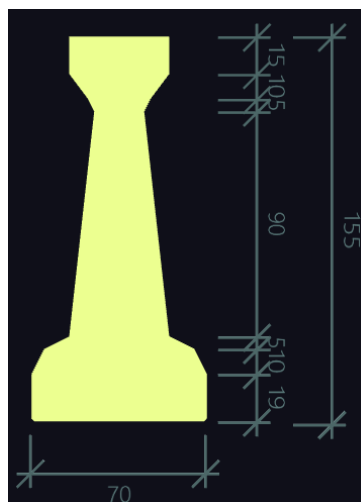
Figura A.1 – Janela para escolha do formato da seção da viga.



(fonte: elaborada pelo autor.)

Para visualizar os valores das dimensões da seção, deve-se clicar em uma das faces transversais da viga na área de visualização 3D do programa, e isso irá aproximar a seção com suas medidas em centímetros, conforme segue:

Figura A.2 – Janela para escolha do formato da seção da viga.



(fonte: elaborada pelo autor.)

Para fechar a visualização da seção deve-se clicar em qualquer área vazia da janela de visualização.

A.2 DEFINIÇÃO DO COMPRIMENTO DA VIGA

Para definir o comprimento da viga deve-se clicar no botão *Size* dentro do grupo *Beam*, e este abrirá um campo de texto para entrada do valor desejado, em centímetros. Após a digitação do valor deve-se pressionar a tecla *Enter* e a representação da viga será atualizada para o tamanho definido.

A.3 DEFINIÇÃO DE PONTOS NA VIGA PARA ENTRADA DE DADOS

Para a inserção de carregamentos ou apoios, ou mesmo pontos específicos para representação do valor de solicitações no visualizador de resultados, é necessário fazer a inserção de *Grids*. Para adicioná-los à viga, deve-se clicar no botão *Add Grid* do grupo *Grid*. A viga será posicionada lateralmente pelo programa para facilitar o posicionamento do *grid* ao longo de seu

comprimento. Com o auxílio do *mouse*, é possível deslocar o *grid* em inserção ao longo da viga, de modo que réguas aparecerão como referência sobre o desenho para a escolha da medida. A precisão nessa função é de 10 cm para cada deslocamento do mouse, podendo ser reposicionado em passos de até 5 mm após a carga inserida, selecionando-se o *grid* e girando o botão de *scroll* do *mouse* com o *grid selecionado*. Uma vez que o *grid* apareça na posição desejada, o botão esquerdo do *mouse* irá adicionar o elemento, permitindo que se continue adicionando outros *grids*, até que o botão direito do *mouse* seja pressionado para encerrar o comando.

A.4 INSERÇÃO DOS CARREGAMENTOS

Os carregamentos são dependentes do tempo, de modo que primeiramente deve-se certificar de que o instante selecionado é o desejado para atribuição da carga. O instante de tempo atual é indicado no canto inferior esquerdo da janela, podendo ser alterado deslizando o seletor do controle com o uso do botão esquerdo do *mouse* clicando ao lado da direção em que se deseja mudar o seletor, ou arrastando-os enquanto mantendo o botão pressionado.

Para inserir um carregamento sobre a viga deve clicar no botão *New Load* do grupo *Loads*. Uma janela de diálogo aparecerá para que o usuário defina a carga que deseja inserir. Para cargas pontuais verticais ou de momento, deve-se marcar a opção na janela de *Nodal Force* e então preencher os campos dentro de *Load Values* de acordo com as indicações em tela. Para cargas distribuídas, deve-se selecionar a opção de *Uniform Load* e definir o valor em *Load Values*. Após definição da carga a ser inserida, deve-se clicar no botão *Insert* para prosseguir ou *Cancel* para abortar o comando. No caso do *Insert*, a janela de diálogo é fechada e uma representação da carga aparecer sobre a viga para ser posicionada com o *mouse*. Para cargas pontuais, um clique com o botão esquerdo do *mouse* irá adicionar a carga, e para cargas distribuídas o primeiro clique definirá o início e o segundo sua posição final. Para cancelar o comando antes de sua conclusão, deve-se pressionar o botão direito do *mouse*.

A.5 CONTROLE DAS IDADES DO CONCRETO

Para inserir, apagar ou visualizar as idades do concreto no programa deve-se clicar no botão *New Point* do grupo *Time Points*. Abrirá então uma janela com um campo para entrada de novas idades, e uma lista de idades já existentes. Para adicionar novas idades a essa lista deve-se digitar o valor desejado em *Concrete Age*, em dias, e então clicar no botão *Add* para inserir. Para remover, deve-se selecionar as idades desejadas da lista exibida e então clicar no botão *Remove*. Para sair da janela, basta clicar no botão *Close*.

Para selecionar a idade atual do concreto deve-se utilizar a barra no canto inferior esquerdo do programa, denominada *Age Slider*, deslocando-se o seletor para a direita ou para a esquerda. Logo a direita da barra aparecerá o valor da idade selecionada.

A.6 INSERÇÃO DE APOIOS

Para inserir apoios na viga deve-se clicar no botão *New Support* do grupo *Support Cond*. Abrirá uma janela segmentada nas três direções possíveis para inserir restrições. Ainda nessa janela existe uma área que exhibe uma vista de como ficará o apoio que está sendo definido. Os campos são indicativos que permitem a inserção de valor de rigidez produzindo molas, ou deslocamentos pré-definidos, ou então apenas marcar a restrição como ideal.

Após configurado o apoio, o botão *Insert* permite posicionar o elemento sobre a viga, sendo concluído com o botão esquerdo do *mouse*, ou cancelado pelo botão direito. Ainda na janela de configuração existem outros dois botões que permitem zerar a configuração clicando em *Reset* ou simplesmente encerrar a janela clicando em *Cancel*.

A.7 DEFINIÇÃO DO CONCRETO

Para escolha do concreto deve-se clicar no botão *Concrete* do grupo *Materials*. As propriedades do concreto são selecionáveis já pré-determinadas, sendo apenas a umidade relativa definida por digitação. Após ajustadas as propriedades desejadas, basta clicar o botão *Ok* para aceitar as novas propriedades ou *Cancel* para manter as propriedades definidas anteriormente.

A.8 VISUALIZANDO RESULTADOS

Para visualizar os resultados das solicitações sobre a estrutura deve-se clicar na aba *Results Analysis* acima da barra de ferramentas. O programa irá verificar se todas as condições para obtenção das solicitações foram satisfeitas e caso contrário irá indicar a necessidade de entrada de informações adicionais. Caso todas as condições mínimas tenham sido atendidas, a área de desenho principal do programa irá mostrar os resultados sobre a estrutura, podendo ser explorados com auxílio do *Mouse*. O *scroll* permite ampliar ou reduzir a imagem, ao passo que o posicionamento do mouse sobre as solicitações indica sua posição e valor. Ainda clicando na aba a esquerda do desenho, em *Results Browser* é possível alterar a visualização das solicitações desabilitando ou habilitando cortante ou momento fletor.

A barra inferior de seleção de idade do concreto, *Age Slider*, permite alterar a visualização dos resultados das solicitações para cada idade do concreto especificada.