

**Modelação de cabos para vigas de betão pré-
esforçado.
Programa de cálculo automático**

Lutonda Adolph Tomalela

Dissertação para obtenção do Grau de Mestre em
Engenharia Civil: Estruturas e Construção
(mestrado integrado)

Orientador: Prof. Doutor João António Saraiva Pires da Fonseca

fevereiro de 2025

Declaração de Integridade

Eu, Lutonda Adolph Tomalela, que abaixo assino, estudante com o número de inscrição 38775 do Mestrado Integrado em Engenharia Civil: Estruturas e Construção da Faculdade de Engenharia, declaro ter desenvolvido o presente trabalho e elaborado o presente texto em total consonância com o **Código de Integridades da Universidade da Beira Interior**.

Mais concretamente afirmo não ter incorrido em qualquer das variedades de Fraude Académica, e que aqui declaro conhecer, que em particular atendi à exigida referenciação de frases, extratos, imagens e outras formas de trabalho intelectual, e assumindo assim na íntegra as responsabilidades da autoria.

Universidade da Beira Interior, Covilhã 26 /02 /2024

(assinatura conforme Cartão de Cidadão ou preferencialmente
assinatura digital no documento original se naquele mesmo formato)

Dedicatória

Aos meus pais, Madalena Diakangua Tomalela e Matondo Tomalela, pelo amor incondicional, pelos ensinamentos e pelo exemplo de força e dedicação. Meus heróis!

Agradecimentos

A Deus, fonte de sabedoria e força, que guia cada passo desta caminhada.

Ao Professor Doutor João Fonseca, meu orientador, pela orientação valiosa, pelas aulas e pelo incentivo ao pensamento crítico.

Ao Fundo Universitário AMI – Assistência Médica Internacional, pelo apoio essencial nos anos letivos 2021/2022, 2022/2023 e 2024/2025, que contribuiu para a concretização deste percurso acadêmico.

À minha família, em especial aos meus pais e irmãos, pelo suporte inabalável, pelos sacrifícios feitos e por sempre acreditarem em mim.

Aos meus amigos, pela amizade genuína, pelo apoio nos momentos desafiadores e pela partilha desta jornada.

Resumo

A presente dissertação centra-se no desenvolvimento de uma ferramenta de cálculo automático, designada TendonTech, concebida para a modelação e análise de cabos em vigas de betão pré-esforçado. O betão pré-esforçado é amplamente utilizado na engenharia estrutural devido à sua capacidade de melhorar o desempenho das estruturas, reduzindo a fissuração e as deformações, além de aumentar a resistência. O comportamento destas estruturas é significativamente influenciado pelas perdas de pré-esforço, cuja quantificação é essencial para garantir a segurança e a eficácia do projeto. Neste contexto, a dissertação enfatiza a avaliação das perdas de pré-esforço, conforme estabelecido na norma NP EN 1992-1-1, que distingue perdas imediatas, como as resultantes do atrito e da deformação elástica, e perdas diferidas, associadas à relaxação do aço, à retração e à fluência do betão. A implementação do TendonTech incorpora a consideração destes fatores, assegurando a conformidade com os requisitos normativos.

A ferramenta foi desenvolvida em Python, integrando o OpenSees para a análise estrutural e a interoperabilidade com software CAD para a modelação geométrica dos cabos. A interface de entrada de dados foi concebida para ser intuitiva, enquanto os resultados incluem saídas detalhadas e um mecanismo de tratamento de erros, garantindo a fiabilidade do programa. A validação do TendonTech foi conduzida através da comparação com os resultados obtidos pelo software ESTICA 2020, Programa de cálculo automático de estruturas de betão pré-esforçado da biblioteca de software de PENCIL Engenharia Lda, evidenciando a coerência do comportamento estrutural modelado. Este trabalho destaca o contributo das ferramentas computacionais para dimensionamento de estruturas pré-esforçadas. O TendonTech reforça a eficiência do processo de projeto, promovendo a aplicação das normas europeias e demonstrando a viabilidade de soluções tecnológicas no domínio da engenharia de estruturas.

Palavras-chave

Betão pré-esforçado;perdas imediatas;perdas diferidas;Diagrama de Magnel;análise estrutural;OpenSeesPy;python;modelação de cabos;TendonTech;cálculo automático

Abstract

This dissertation focuses on the development of an automated calculation tool, named TendonTech, designed for the modeling and analysis of tendons in prestressed concrete beams. Prestressed concrete is widely utilized in structural engineering due to its ability to enhance structural performance by reducing cracking and deformation, as well as increasing strength. The behavior of these structures is significantly influenced by prestress losses, the quantification of which is essential to ensure the safety and effectiveness of the design. In this context, the dissertation emphasizes the evaluation of prestress losses as outlined in the NP EN 1992-1-1 standard, which distinguishes between immediate losses, such as those resulting from friction and elastic deformation, and deferred losses associated with steel relaxation, concrete shrinkage, and creep. The implementation of TendonTech incorporates consideration of these factors, ensuring compliance with regulatory requirements.

The tool was developed in Python, integrating OpenSees for structural analysis and interoperability with CAD software for the geometric modeling of tendons. The data input interface was designed to be intuitive, while the outputs include detailed results and an error handling mechanism, ensuring the program's reliability. Validation of TendonTech was conducted through comparison with results obtained from ESTICA 2020, an automated calculation program for prestressed concrete structures from the software library of PENCIL Engenharia Lda, demonstrating the consistency of the modeled structural behavior. This work highlights the contribution of computational tools to the design of prestressed structures. TendonTech enhances the efficiency of the design process, promotes the application of European standards, and demonstrates the feasibility of technological solutions in the field of structural engineering.

Keywords

Prestressed concrete; immediate losses; time dependant losses; Magnel diagram; structural analysis; OpenSeesPy; Python; tendon modeling; TendonTech; automated calculation

Índice

CAPÍTULO 1	1
1. INTRODUÇÃO	1
1.1. CONTEXTUALIZAÇÃO DO TEMA	1
1.2. OBJETIVOS	3
1.3. ESTRUTURA DA DISSERTAÇÃO	4
CAPÍTULO 2	5
2. FUNDAMENTOS TEÓRICOS	5
2.1. PRÉ-ESFORÇO	5
2.1.2. <i>Técnicas e sistemas pré-esforço</i>	6
2.2. CONCEITOS DE PRÉ-ESFORÇO	7
2.2.1. <i>Estado de tensão</i>	7
2.2.2. <i>Resultante das compressões “C”</i>	8
2.2.3. <i>Carga equivalente</i>	9
2.3. MODELAÇÃO DE CABOS DE PRÉ-ESFORÇO	11
2.3.1. <i>Geometrias práticas e equações que definem os traçados</i>	11
2.4. ELEMENTOS UTILIZADOS NO PRÉ-ESFORÇO	13
2.4.1. <i>Betão estrutural</i>	13
2.4.2. <i>Aço de pré-esforço</i>	14
2.4.3. <i>Bainhas e ancoragens</i>	15
2.5. PERDAS DE PRÉ-ESFORÇO	16
2.5.1. <i>Força de tensionamento</i>	17
2.6. PERDAS IMEDIATAS DE PRÉ-ESFORÇO	17
2.6.1. <i>Perdas devidas ao atrito</i>	17
2.6.2. <i>Perdas devidas à reentrada das cunhas nas ancoragens</i>	18
2.6.3. <i>Perdas devidas à deformação instantânea do betão</i>	19
2.7. PERDAS DIFERIDAS DE PRÉ-ESFORÇO	20
CAPÍTULO 3	21
3. DESENVOLVIMENTO DO PROGRAMA DE CÁLCULO AUTOMÁTICO	21
3.1. REQUISITOS E ESPECIFICAÇÕES DO PROGRAMA	21

3.1.1.	<i>Funcionalidades</i>	21
3.1.2.	<i>Limitações</i>	21
3.2.	INTERAÇÃO UTILIZADOR - PROGRAMA	22
3.2.1.	<i>Interface e fluxo de trabalho</i>	22
3.2.2.	<i>Entrada de dados</i>	23
3.2.3.	<i>Interação utilizador - programa</i>	24
3.2.4.	<i>Mensagens e tratamento de erros</i>	24
3.2.5.	<i>Documentação e suporte</i>	25
3.3.	LIGUAGEM DE PROGRAMAÇÃO: PYTHON	25
3.4.	IMPLEMENTAÇÃO DO PROGRAMA	26
3.4.1.	<i>Definição geométrica da viga</i>	26
3.4.2.	<i>Análise estrutural com OpenSeesPy: Análise estática, Diagrama de Magnel e Força de tensionamento</i>	28
3.4.3.	<i>Modelação geométrica dos cabos</i>	49
3.4.4.	<i>Quantificação das perdas imediatas</i>	55
3.4.5.	<i>Quantificação das perdas diferidas</i>	67
3.4.6.	<i>Relatório de cálculo</i>	74
CAPÍTULO 4		82
4. APLICAÇÃO PRÁTICA DO PROGRAMA DE CÁLCULO AUTOMÁTICO		82
4.1.	DESCRIÇÃO DO ESTUDO DE CASO	82
4.2.	APLICAÇÃO DO PROGRAMA	82
4.3.	ANÁLISE E DISCUSSÃO DOS RESULTADOS	85
4.4.	VALIDAÇÃO DO PROGRAMA	91
CAPÍTULO 5		94
5. CONCLUSÕES E TRABALHOS FUTUROS		94
5.1.	CONCLUSÕES	94
5.2.	RECOMENDAÇÕES PARA TRABALHOS FUTUROS	95
REFERÊNCIAS		96
ANEXO 1		98
A. RELATÓRIO DE CÁLCULO: APLICAÇÃO PRÁTICA DO PROGRAMA		98
RELATÓRIO DE CÁLCULO		99

CONVENÇÃO DE SINAIS	99
CARATERÍSTICAS DO BETÃO	99
PROPRIEDADES GEOMÉTRICAS DA SECÇÃO TRANSVERSAL DE BETÃO	99
CARATERÍSTICAS DO SISTEMA DE PRÉ-ESFORÇO.....	100
CARATERÍSTICAS DO AÇO DE PRÉ-ESFORÇO	100
CARATERÍSTICAS DOS CABOS DE PRÉ-ESFORÇO E FORÇA DE TENSIONAMENTO	100
CARGAS E FATORES DE REDUÇÃO	100
PERDAS INICIAIS DE PRÉ-ESFORÇO	100
CABO I: PRÉ-ESFORÇO APÓS PERDAS POR ATRITO, PM	100
CABO I: PRÉ-ESFORÇO APÓS PERDAS POR REENTRADA DAS CUNHAS, PSL.....	103
CABO I: PRÉ-ESFORÇO APÓS PERDAS POR DEFORMAÇÃO ELÁSTICA DO BETÃO, PEL	105
CABO II: PRÉ-ESFORÇO APÓS PERDAS POR ATRITO, PM	108
CABO II: PRÉ-ESFORÇO APÓS PERDAS POR REENTRADA DAS CUNHAS, PSL	110
PRÉ-ESFORÇO APÓS PERDAS INSTANTÂNEAS, PM0	113
PRÉ-ESFORÇO APÓS PERDAS DIFERIDAS, P ∞	118
PRÉ-ESFORÇO FINAL.....	123
ANEXOS.....	126
<i>Pré-esforço útil mínimo - Vãos</i>	126
<i>Pré-esforço útil mínimo - Apoio</i>	127
<i>Variação do pré-esforço</i>	128
<i>Equações que definem o traçado dos cabos</i>	129
ANEXO 2	130
B. MANUAL DO UTILIZADOR.....	130
INTRODUÇÃO	130
REQUISITOS DE SISTEMA.....	130
INSTALAÇÃO	130
UTILIZAÇÃO – ENTRADA DE DADOS	134
UTILIZAÇÃO – TENDONTECH.....	137
ERROS	141
CONSIDERAÇÕES FINAIS.....	142

Lista de Figuras

Figura 1.1 - (a) Viga simplesmente armada e secção transversal. (b) Diagrama de corpo livre, distribuição de tensões e forças resultantes [1] (adaptado).	1
Figura 1.2 - (a) Pré-esforço reto numa viga curva. (b) Distribuição de tensões nas secções a meio vão e nos apoios, respetivamente [2].	2
Figura 1.3 - Exemplos típicos do perfil dos cabos: (a) traçado parabólico para uma situação de carregamento uniformemente distribuído; (b) traçados retos para uma situação de carregamento pontual; (c) diagrama de tensões normais a meio vão para as situações (a) e (b) [3] (adaptado).	2
Figura 2.1 – Fases da aplicação de pré-esforço por pré-tensão [7].	6
Figura 2.2 – Aplicação de pré-esforço por pós-tensão. [7] (adaptado).	7
Figura 2.3 – Viga simplesmente apoiada sob uma carga uniformemente distribuída (a); Estado de tensão na viga devido aos esforços internos (b) [7] (adaptado).	8
Figura 2.4 – Comparação entre os diagramas de corpo livre de uma viga simplesmente apoiada simplesmente armada e uma viga simplesmente apoiada pré-esforçada; (a) Viga de betão armado sem carregamento; (b) Viga pré-esforçada sem carregamento; (c) Viga de betão armado com um carregamento w_1 ; (d) Viga pré-esforçada com um carregamento w_1 ; (e) Viga de betão armado com um carregamento corrente w ; (f) Viga pré-esforçada com um carregamento corrente w [2].	9
Figura 2.5 – Cargas equivalentes de pré-esforço; (a) Cabo com uma mudança de declive abrupta; (b) Cabo parabólico. [2].	10
Figura 2.6 – Cargas Equivalentes de pré-esforço; (a) Carga distribuída; (b) Carga pontual à meio vão; (c) Duas cargas pontuais a distâncias de aL e $L - aL$ dos apoios, respetivamente [7] (adaptado).	11
Figura 2.7 – Viga simplesmente apoiada com cabo de pré-esforço de traçado parabólico, onde y_m representa a flecha, e c.g.c. corresponde ao eixo neutro [8].	11
Figura 2.8 – Viga contínua com traçados parabólicos [7].	12
Figura 2.9 - Perfil de um segmento parabólico [7] (adaptado).	12
Figura 2.10 – (a) Relação tensão-extensão para o betão cintado e não cintado; (b) Influência da fluência no betão comprimido [5], [11] (adaptado).	13
Figura 2.11 – Diagramas típicos de tensão-extensão de aços de pré-esforço [1].	15
Figura 2.12 - Tipos de ancoragens. Ancoragem ativa (a). Ancoragem passiva (b). Dispositivo de acoplamento (c) [14] (adaptado).	16
Figura 2.13 - Tipos de perdas de pré-esforço [7].	17

Figura 2.14 - Gráfico típico da variação do pré-esforço após as perdas por atrito numa viga simplesmente apoiada [7] (adaptado).....	18
Figura 2.15 - Gráfico típico da variação do pré-esforço após as perdas por reentrada das cunhas numa viga simplesmente apoiada [7] (adaptado).	19
Figura 3.1 – Interação utilizador – programa.....	23
Figura 3.2 – Entrada de dados.....	23
Figura 3.3 - Erro comum à entrada dos dados.....	24
Figura 3.4 - Erro complexo. Dependência externa OpenSeesPy.	25
Figura 3.5 - Viga contínua com 3 tramos.....	26
Figura 3.6 - Secção transversal em “T”.	26
Figura 3.7 - Convenção de sinais.....	27
Figura 3.8 - Processo construtivo: Cimbra ao solo.	32
Figura 3.9 - Cimbra ao solo, fase 1: diagrama de momentos fletores.	36
Figura 3.10 – Cimbra ao solo, fase 2: diagrama de momentos fletores.	36
Figura 3.11 - Cimbra ao solo, fase 3: diagrama de momentos fletores.....	36
Figura 3.12 - Bases para as condições de tensão [3].	37
Figura 3.13 - Diagrama de Magnel [3] (adaptado).....	38
Figura 3.14 - Modelo estático da viga.....	42
Figura 3.15 - Diagrama de Magnel: pré-esforço útil mínimo nos vãos.....	46
Figura 3.16 - Diagrama de Magnel: pré-esforço útil mínimo nos apoios interiores.....	48
Figura 3.17 - Resumo dos parâmetros de pré-esforço determinados pelo diagrama de Magnel.....	49
Figura 3.18 – Visualização do traçado dos cabos em CAD, para uma viga contínua com dois vãos.	55
Figura 3.19 - Perdas de pré-esforço por atrito.	58
Figura 3.20 - Variação das perdas de pré-esforço por ao atrito.....	58
Figura 3.21 – Variação das perdas devidas a reentrada das cunhas nas ancoragens.	61
Figura 3.22 - Perdas devidas à reentrada das cunhas nas ancoragens e alongamentos teóricos.	61
Figura 3.23 - Cargas equivalentes ao pré-esforço; Diagrama de momentos fletores associados.....	64
Figura 3.24 - Perdas devidas à deformação elástica do betão.	66
Figura 3.25 - Variação das perdas devidas à deformação elástica do betão.	66
Figura 3.26 - Pré-esforço após as perdas diferidas.	72
Figura 3.27 - Variação do pré-esforço após perdas diferidas.....	73
Figura 3.28 - Variação total do pré-esforço.	74
Figura 4.1 - Viga contínua. Exemplo de aplicação prática.	82

Figura 4.2 - TendonTech: Definição de cargas e características da viga. Viga contínua com dois tramos iguais de 33 m e carregamento uniformemente distribuído de 31 kN/m.	83
Figura 4.3 - TendonTech: Configuração da secção transversal da viga. Secção transversal em “T”.	83
Figura 4.4 - TendonTech: Definição dos materiais. Betão C30/37 e aço de pré-esforço Y1860.	84
Figura 4.5 - TendonTech: Definição do perfil dos cabos. Número de cabos e número de tensionamentos especificados.	85
Figura 4.6 - TendonTech: Solução inicial. Número de cabos, cordões e força de tensionamento.	85
Figura 4.7 - TendonTech: Perdas por atrito nos cabos I e II.	86
Figura 4.8 - TendonTech: Variação das perdas por atrito nos cabos I e II.	86
Figura 4.9 - TendonTech: Perdas devidas a reentrada das cunhas para os cabos I e II. Alongamentos teóricos no primeiro e segundo tensionamentos para os cabos I e II. ...	87
Figura 4.10 - TendonTech: Variação das perdas por reentrada das cunhas, nos cabos I e II.	87
Figura 4.11 - TendonTech: Pré-esforço após perdas imediatas no Cabo I. Tensões nas secções críticas ao longo do comprimento da viga. Extensões.	88
Figura 4.12 - TendonTech: Perdas devidas à deformação elástica do betão no Cabo I.	88
Figura 4.13 - TendonTech: Variação das perdas devidas a deformação elástica do betão no Cabo I.	89
Figura 4.14 - TendonTech: Pré-esforço após perdas imediatas no Cabo II. Tensões nas secções críticas ao longo do comprimento da viga. Extensões.	89
Figura 4.15 - TendonTech: Pré-esforço após as perdas diferidas nos cabos I e II.	90
Figura 4.16 - TendonTech: Pré-esforço final. Pré-esforço após as perdas imediatas e pré-esforço após as perdas diferidas.	90
Figura 4.17 - TendonTech: Variação final do pré-esforço.	90

Lista de Tabelas

Tabela 2.1 - Vantagens da utilização do pré-esforço [7] (adaptado).	5
Tabela 2.2 – Valores característicos da resistência à compressão do betão [9] [5]. (adaptado).....	13
Tabela 2.3 – Características dos aços de pré-esforço [1], [13]. (adaptado).....	15
Tabela 2.4 - Coeficientes de atrito típicos para diferentes tipos de aços de pré-esforço e bainhas [9].	18
Tabela 3.1 – Criação de um ambiente virtual para isolamento das dependências externas através da linha de comandos do Windows.....	22
Tabela 3.2 - Propriedades geométricas da secção transversal em “T”.	27
Tabela 3.3 – Código para importação de dados de entrada do TendonTech: Geometria.	28
Tabela 3.4 – Código para importação de dados de entrada do TendonTech: Análise estática	29
Tabela 3.5 – Código para a divisão do comprimento total da viga em intervalos de até 1 metro.....	31
Tabela 3.6 – Código para simulação do processo construtivo: Cimbra ao solo	32
Tabela 3.7 - Inequações de Magnel.....	37
Tabela 3.8 - Código para executar a combinação de ações.....	38
Tabela 3.9 - Código para importação de dados de entrada do TendonTech: Diagrama de Magnel.	42
Tabela 3.10 - Código para a construção do diagrama de Magnel na secção critica onde o momento fletor é positivo.....	44
Tabela 3.11 - Código para a construção do diagrama de Magnel na secção critica onde o momento fletor é negativo.	46
Tabela 3.12 - Código necessário para a definição da força de tensionamento.	48
Tabela 3.13 – Código necessário para a definição das equações que definem o traçado dos cabos.	49
Tabela 3.14 – Código necessário para a visualização da geometria dos cabos no caso de uma viga simples.	52
Tabela 3.15 – Código necessário para a visualização do traçado dos cabos no caso de uma viga contínua com dois vãos.	53
Tabela 3.16 - Código necessário para calcular a soma dos desvios angulares ao longo do comprimento da viga.	55
Tabela 3.17 - Código para quantificar as perdas de pré-esforço por atrito.	56

Tabela 3.18 - Código necessário para estimar as perdas devidas a reentrada das cunhas nas ancoragens.	59
Tabela 3.19 - Código necessário para estimar os esforços equivalentes ao pré-esforço na viga.	61
Tabela 3.20 - Código necessário para calcular as perdas devidas ao encurtamento elástico do betão.	64
Tabela 3.21 – Código necessário para determinar o coeficiente de fluência.	67
Tabela 3.22 – Código necessário para determinar a extensão de retração.....	67
Tabela 3.23 – Código necessário para determinar a relaxação do aço.	69
Tabela 3.24 – Código necessário para o cálculo das perdas diferidas.	70
Tabela 3.25 - Código necessário para determinar o pré-esforço final.	73
Tabela 3.26 - Código necessário para gerar o relatório de cálculo.	74
Tabela 4.1 – Validação dos resultados: Parâmetros calculados.....	91
Tabela 4.2 – Validação dos resultados: alongamentos teóricos.	92
Tabela 4.3 - Validação dos resultados: Pré-esforço após as perdas imediatas e pré-esforço útil.	92
Tabela 4.4 - Validação dos resultados: tensões no betão devidas ao peso próprio e ao pré-esforço.	92

Lista de Acrónimos

NSC	High Strength Concrete
HSC	Normal Strength Concrete
ETA	European Technical Assessment
ELS	Estados Limites de Serviço
ELU	Estados Limites Últimos

Capítulo 1

1. Introdução

1.1. Contextualização do tema

O betão armado é o material estrutural mais utilizado desde o final do século XIX e deverá manter essa posição no futuro próximo [1]. O betão possui uma elevada resistência à compressão, mas apresenta uma fraca resistência à tração. A sua resistência à tração varia entre 8 e 14 por cento da sua resistência à compressão [2]. Devido à baixa resistência à tração do betão, as fissuras por flexão em vigas desenvolvem-se em fases iniciais de carregamento [2], assim, varões de aço são incorporados ao seu interior para resistir às tensões de tração resultantes de cargas aplicadas, deformações ou efeitos independentes das cargas, como variações de temperatura e retração. [1].

Considere-se a viga simplesmente armada ilustrada na Figura 1.1, onde as cargas externas provocam tensões de tração na fibra inferior da viga, resultando em fissuração [1]. As vigas de betão armado, na prática, tendem a apresentar fissuração sob as ações correspondentes aos Estados Limites de Serviço (ELS) [1].

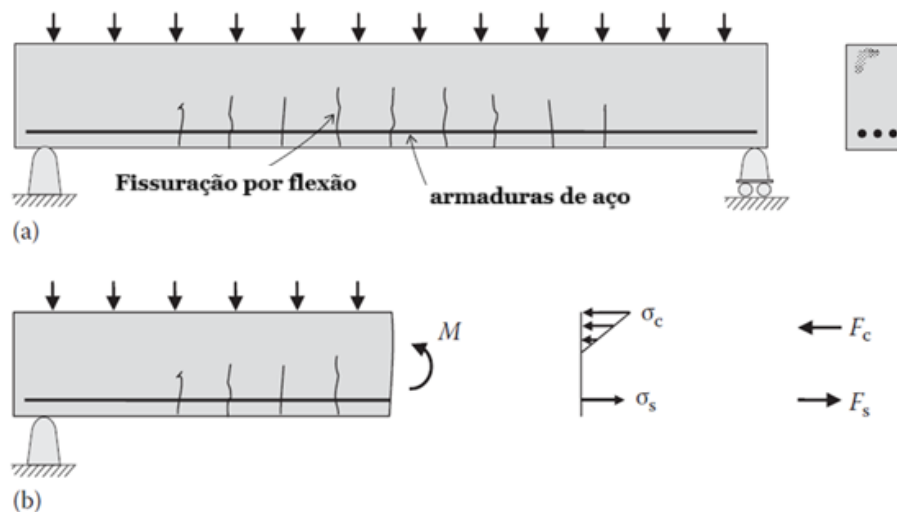


Figura 1.1 - (a) Viga simplesmente armada e secção transversal. (b) Diagrama de corpo livre, distribuição de tensões e forças resultantes [1] (adaptado).

Para reduzir ou prevenir o desenvolvimento das fissuras ilustradas na Figura 1.1, impõe-se uma força concêntrica ou excêntrica na direção longitudinal da viga, a qual elimina ou reduz significativamente as tensões de tração nas secções críticas aumentando assim as capacidades de flexão, esforço transversal e torção das secções, como ilustrado na Figura 1.2 [2].

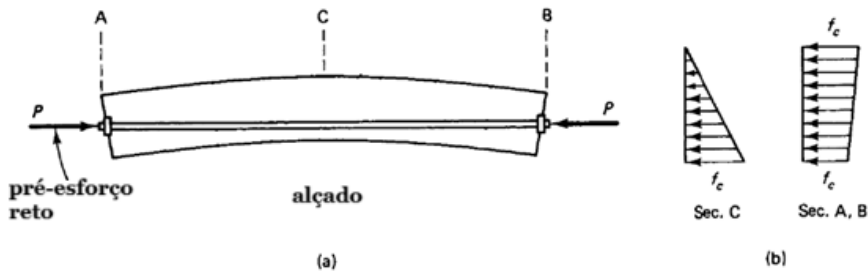


Figura 1.2 - (a) Pré-esforço reto numa viga curva. (b) Distribuição de tensões nas secções a meio vão e nos apoios, respetivamente [2].

A força P , ilustrada na Figura 1.2, é denominada força de pré-esforço, que consiste numa força de compressão que comprime as secções ao longo do vão do elemento estrutural antes da aplicação das restantes cargas permanentes. A escolha do tipo e da magnitude da força de pré-esforço é determinada principalmente com base no tipo de sistema a ser construído, no comprimento do vão e na esbelteza desejada [2]. Como a força de pré-esforço é aplicada ao longo do eixo do elemento, o princípio envolvido é comumente conhecido como pré-esforço linear [2].

O traçado dos cabos de pré-esforço é normalmente definido e calculado com base no diagrama de momentos fletores da viga, como ilustrado na Figura 1.3. A complexidade do traçado pode variar consoante a situação; em vigas com vãos muito longos, além de treços retos, parabólicos ou circulares em perfil, podem também existir geometrias semelhantes em planta. Estas geometrias não lineares tornam o cálculo e a representação dos cabos um processo complexo que requer ferramentas adequadas, pois as equações que definem o traçado dos cabos afetam diretamente a quantificação das perdas de pré-esforço.

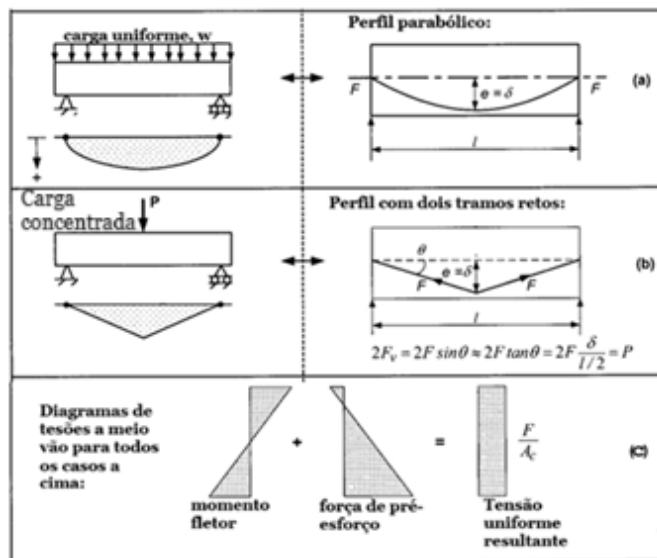


Figura 1.3 - Exemplos típicos do perfil dos cabos: (a) traçado parabólico para uma situação de carregamento uniformemente distribuído; (b) traçados retos para uma situação de carregamento pontual; (c) diagrama de tensões normais a meio vão para as situações (a) e (b) [3] (adaptado).

O dimensionamento preliminar de vigas de betão pré-esforçado é normalmente realizado com base nas limitações das tensões de serviço [4]. Para facilitar o trabalho, foram desenvolvidas várias técnicas que agilizam a otimização da secção transversal da viga, da força de pré-esforço e da excentricidade dos cabos [4]. Um dos primeiros e mais reconhecidos auxiliares de projeto é o diagrama de Magnel, criado em 1948 [4], que será amplamente usado neste trabalho. Este diagrama permite definir, para uma secção crítica, as combinações adequadas de excentricidade e força de pré-esforço.

Destaca-se a importância de desenvolver um programa de cálculo automático, acessível e intuitivo, que otimiza o tempo, reduz erros humanos e assegura a conformidade do projeto de aplicação de pré-esforço com a NP EN 1992-1-1 [5].

No contexto académico, esta ferramenta poderá ser um recurso valioso para uma melhor compreensão dos princípios do pré-esforço; no âmbito profissional, pode funcionar como uma ferramenta prática que apoia o dimensionamento inicial dos cabos e a quantificação das perdas de pré-esforço em estruturas complexas.

1.2. Objetivos

O objetivo principal deste trabalho é desenvolver um programa de cálculo automático que apoie o projeto de estruturas de betão pré-esforçado com cabos de traçado complexo. Para alcançar esse objetivo, será desenvolvida uma ferramenta computacional de fácil utilização, destinada a ser empregue em unidades curriculares do curso de Mestrado em Engenharia Civil. O programa será concebido para ser testado e, posteriormente, divulgado para um uso mais amplo.

Este trabalho envolve a modelação do traçado geométrico espacial de cabos de pré-esforço, utilizando parâmetros e equações matemáticas, com o objetivo de obter curvas de derivada contínua, compostas por sequências de troços retos, parabólicos ou circulares. A representação dos cabos será realizada através de um programa de CAD corrente, e a linguagem de programação Python será utilizada para o desenvolvimento do software.

Adicionalmente, o programa incluirá a análise automática da estrutura sob a ação do tensionamento dos cabos. Será programada a entrada de dados e a saída de resultados para a análise estrutural, bem como a quantificação das perdas de pré-esforço com base na NP EN 1992-1-1 [5].

O objetivo final é desenvolver uma ferramenta computacional que simplifique e otimize o processo de projeto de estruturas de betão pré-esforçado. Ao integrar modelação geométrica, análise estrutural e cálculo das perdas de pré-esforço em um único programa, o trabalho visa proporcionar a estudantes e profissionais de engenharia civil uma solução prática para o dimensionamento de vigas pré-esforçadas com cabos de traçado complexo.

1.3. Estrutura da dissertação

A dissertação encontra-se organizada em cinco capítulos principais, conforme a seguinte estrutura:

- Capítulo 2: Fundamentos teóricos

Este capítulo apresenta, de forma simplificada, os conceitos fundamentais essenciais para a compreensão do tema. Serão explorados os princípios básicos do betão pré-esforçado, as suas vantagens e aplicações. Além disso, serão discutidos os métodos de modelação dos cabos de pré-esforço e as perdas associadas, com base na NP EN 1992-1-1 [5].

- Capítulo 3: Desenvolvimento do programa de cálculo automático

Este capítulo descreve o processo de desenvolvimento do programa de cálculo automático, englobando os requisitos e especificações iniciais, a seleção da linguagem de programação e a implementação das respetivas funcionalidades. Adicionalmente, aborda a integração com software CAD para a visualização dos cabos e a análise estrutural. Por fim, são apresentados os procedimentos de entrada de dados, geração de resultados e validação do programa, por meio da comparação dos resultados obtidos com o software ESTICA 2020 [6].

- Capítulo 4: Aplicação prática do programa de cálculo automático

Este capítulo apresenta uma aplicação prática do programa, baseada num exemplo analisado na unidade curricular (10304/15298) Betão Pré-Esforçado. São descritos o processo de aplicação e os resultados obtidos, seguidos de uma análise crítica.

- Capítulo 5: Conclusões e trabalhos futuros

No capítulo final, são apresentadas as conclusões do trabalho realizado, discutindo as limitações encontradas durante o desenvolvimento e a aplicação do programa. Por fim, são apresentadas recomendações para trabalhos futuros, com vista à expansão e/ou aperfeiçoamento do programa desenvolvido.

Capítulo 2

2. Fundamentos teóricos

2.1. Pré-esforço

O pré-esforço é um princípio [3]. O conceito original, promovido por Freyssinet, procurava eliminar completamente as tensões de tração, criando assim um material sem fissuras sob as cargas de serviço. Contudo, este conceito evoluiu para contrariar apenas parcialmente as tensões de tração induzidas externamente, permitindo uma tensão e fissuração controladas, de forma análoga ao betão armado, o que levou ao desenvolvimento do que hoje se designa como pré-esforço parcial [3]. O betão parcialmente pré-esforçado combina elementos do betão armado e do betão totalmente pré-esforçado, situando-se entre ambos [3]. Atualmente, é difícil discutir qualquer um dos materiais sem considerar a sua combinação, uma vez que ambos utilizam essencialmente os mesmos componentes, aço e betão, e o seu desenvolvimento histórico está interligado [3].

2.1.1. Vantagens e aplicações do pré-esforço

O betão pré-esforçado apresenta várias vantagens em comparação com o betão armado, uma vez que os elementos de betão totalmente pré-esforçados tendem a estar sujeitos a compressão ao longo da sua vida útil, corrigindo assim várias limitações do betão [7]. Apresenta-se na Tabela 2.1 as vantagens de um elemento pré-esforçado em comparação com um elemento equivalente de betão armado; para cada efeito, são listados os benefícios.

Tabela 2.1 - Vantagens da utilização do pré-esforço [7] (adaptado).

Vantagens	Detalhes
1) Controlo da fissuração sob as cargas de serviço	<ul style="list-style-type: none">- Redução da corrosão do aço;- Aumento da durabilidade, mais rigidez e menos deformações;- Adequado para uso em estruturas de retenção de fluídos;- Melhor desempenho sob cargas dinâmicas e fadiga.
2) Relação vão-altura	<ul style="list-style-type: none">- Vãos mais longos;- Redução do peso próprio;- Secções mais económicas.
3) Adequado para soluções pré-fabricadas	<ul style="list-style-type: none">- Construção rápida;- Melhor controlo de qualidade;- Adequado para construções repetitivas;- Redução do uso de cofragens;- Disponibilidade de secções normalizadas.

Embora o pré-esforço apresente várias vantagens, é imprescindível considerar certos aspectos críticos. A aplicação do pré-esforço exige tecnologia avançada e mão-de-obra especializada, e a utilização de materiais de alta resistência pode resultar em custos elevados. [7].

2.1.2. Técnicas e sistemas pré-esforço

Existem vários métodos e técnicas de pré-esforço disponíveis; no entanto, exceto pelo pré-esforço químico, a maioria pode ser classificada em dois grupos principais: pré-tensão e pós-tensão [3]. Alguns métodos são especificamente associados a uma aplicação particular, mas ainda assim pertencem a um dos grupos mencionados [3].

Na pré-tensão, os cabos de pré-esforço são esticados até atingirem uma tensão predeterminada e ancorados aos moldes, sendo posteriormente betonados [3]. Após a cura do betão, os cabos são libertados e, devido ao atrito na ligação entre os cabos e o betão, este fica comprimido [3]. Assim, o prefixo 'pré' em pré-tensão refere-se ao facto de os cabos serem tensionados antes do endurecimento do betão [3]. As várias fases da operação de pré-tensão são resumidas da seguinte forma: ancoragem dos cabos contra os encontros de extremidade, colocação dos macacos, aplicação de tensão aos cabos, betonagem e corte dos cabos [7], como ilustrado na Figura 2.1.

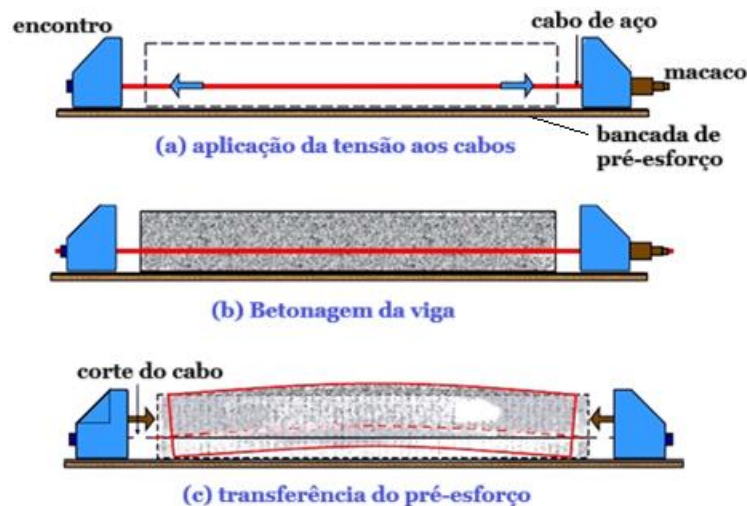


Figura 2.1 – Fases da aplicação de pré-esforço por pré-tensão [7].

Na pós-tensão, os cabos são tensionados e ancorados nas extremidades do elemento de betão após este ter sido betonado e atingido resistência suficiente, e normalmente, uma bainha metálica é colocada ao longo do elemento antes da betonagem [3]. As armaduras podem ser posicionadas soltas dentro da bainha antes da betonagem ou inseridas após o endurecimento do betão [3]. Após o tensionamento, o espaço entre as armaduras e a bainha é preenchido com calda de cimento, que endurece, assegurando a aderência das armaduras ao betão, melhorando a resistência à fissuração e reduzindo os riscos de corrosão [3].

Na técnica de pós-tensão com armaduras não aderentes, a bainha é preenchida com massa lubrificante, eliminando a aderência, com a força aplicada apenas nas ancoragens [3]. Estas armaduras, revestidas com massa lubrificante, são geralmente utilizadas em sistemas de lajes de estruturas residenciais e de estacionamento com vários vãos, devido à sua eficiência e economia [3]. Em aplicações de pré-esforço exterior, as armaduras são também não aderentes, mas colocadas fora da secção de betão [3].

Apresenta-se, na Figura 2.2 uma representação esquemática de um elemento pós-tensionado, onde perfil do cabo varia conforme as condições de apoio [7]. Numa viga simplesmente apoiada, a o cabo segue um perfil côncavo entre as extremidades, e, numa viga contínua, o cabo é côncavo nos vãos e convexo sobre os apoios [7].

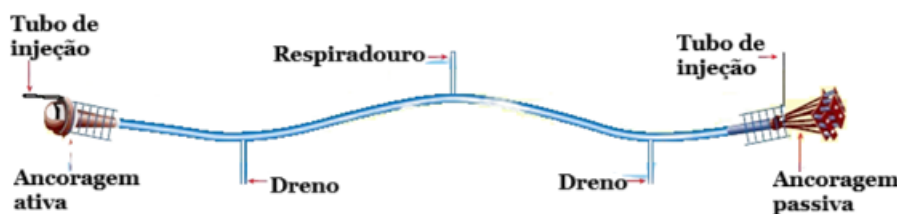


Figura 2.2 – Aplicação de pré-esforço por pós-tensão. [7] (adaptado).

2.2. Conceitos de pré-esforço

As análises do pré-esforço na transferência e sob as cargas de serviço são realizadas em conjunto, e, nestas condições, o aumento da tensão no aço devido à flexão é desconsiderado. Existem três conceitos principais para analisar um elemento pré-esforçado, baseadas em conceitos específicos, que são [7]:

- Estado de Tensão;
- Resultante das Compressões “C”;
- Carga Equivalente.

2.2.1. Estado de tensão

Em elementos de betão pré-esforçado, as tensões nas fibras de são diretamente calculadas a partir do pré-esforço longitudinal aplicado e pela carga externa [2], como indicado na equação (1).

$$\sigma = -\frac{P}{A} \pm \frac{P \cdot e \cdot y}{I} \pm \frac{M \cdot y}{I} \quad (1)$$

Onde: P é a força de pré-esforço, A é a área da secção transversal do betão, e é a excentricidade do cabo de pré-esforço, y é a distância entre o centro geométrico da secção transversal do betão e a fibra mais tracionada ou comprimida, e I é o momento de inércia da secção.

O conceito de estado de tensão é utilizado para comparar as tensões calculadas com as tensões admissíveis [7]. Considere a Figura 1.3, que ilustra uma viga simplesmente apoiada, sujeita a uma carga uniformemente distribuída e a um pré-esforço com excentricidade constante ao longo do seu comprimento; o estado de tensão para esta condição, mostrando no primeiro perfil a tensão devida ao pré-esforço, no segundo perfil a tensão resultante da excentricidade da força de pré-esforço, e no terceiro perfil a tensão provocada pelo momento fletor devido ao peso próprio [7].

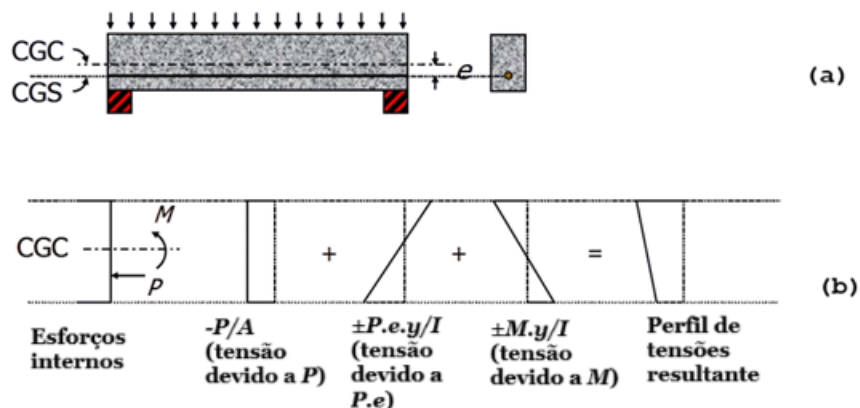


Figura 2.3 – Viga simplesmente apoiada sob uma carga uniformemente distribuída (a); Estado de tensão na viga devido aos esforços internos (b) [7] (adaptado).

2.2.2. Resultante das compressões “C”

A abordagem baseada no conceito de resultante das compressões é análoga ao estudo do betão armado, onde a tensão no aço de pré-esforço e a compressão resultante no betão são consideradas para equilibrar as cargas externas [7]. Esta abordagem é utilizada para determinar as dimensões de uma secção e verificar a capacidade de carga em serviço [7]. Por outro lado, este mesmo conceito analisa a viga como se fosse uma viga elástica de betão simples, utilizando os princípios básicos da estática, considerando a força de pré-esforço como uma força de compressão externa, com uma força de tração constante no cabo ao longo do vão [2]. Ambas as abordagens resultam no cálculo das mesmas tensões no betão, que são então comparadas com as tensões admissíveis nas extremidades [7].

A resultante das compressões no betão e a força de tração no aço formam em conjunto um binário, cuja magnitude é igual ao momento fletor aplicado e é calculada com ilustrado na Figura 2.4 [1].

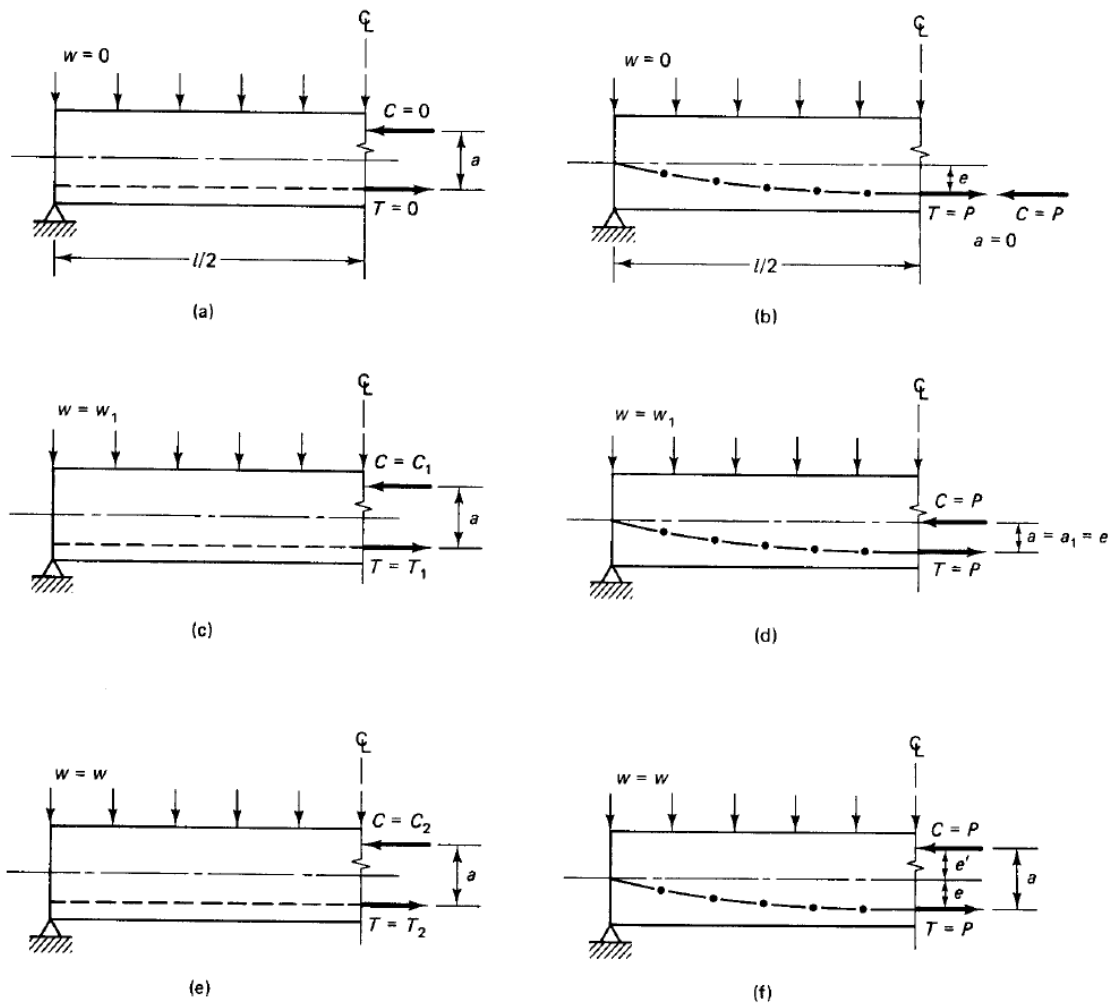


Figura 2.4 – Comparação entre os diagramas de corpo livre de uma viga simplesmente apoiada simplesmente armada e uma viga simplesmente apoiada pré-esforçada; (a) Viga de betão armado sem carregamento; (b) Viga pré-esforçada sem carregamento; (c) Viga de betão armado com um carregamento w_1 ; (d) Viga pré-esforçada com um carregamento w_1 ; (e) Viga de betão armado com um carregamento corrente w ; (f) Viga pré-esforçada com um carregamento corrente w [2].

Note-se que, em estados limites últimos (ELU), as vigas de betão armado e as vigas pré-esforçadas não se distinguem.

2.2.3. Carga equivalente

Os efeitos de um cabo de pré-esforço numa viga podem ser substituídos pelos efeitos de cargas equivalentes, onde estas cargas podem ser facilmente identificadas ao desenhar o diagrama de corpo livre do betão sem o aço [3]. As cargas equivalentes incluem forças horizontais e verticais concentradas, momentos nos apoios de extremidade e forças transversais ao longo do perfil do cabo [3]. As forças transversais podem manifestar-se sob a forma de uma força concentrada devido a uma mudança abrupta na inclinação do perfil do cabo, uma carga uniforme ou uma carga distribuída variável [3].

Esta técnica baseia-se na utilização das cargas equivalentes de pré-esforço para equilibrar ou contrabalançar a carga gravítica imposta à qual uma viga está sujeita [2], como ilustrado na Figura 2.5.

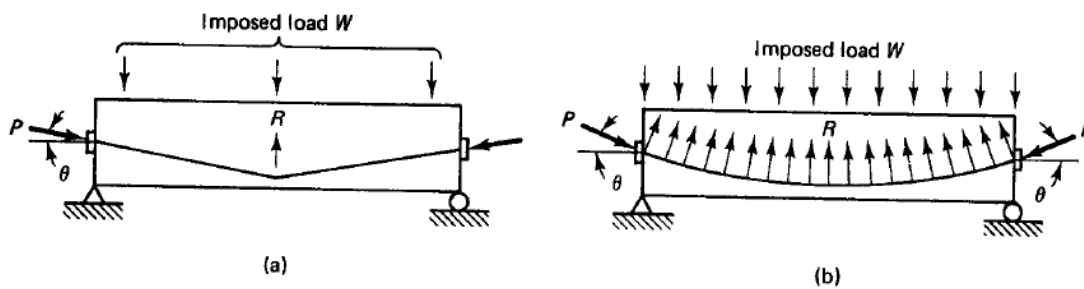


Figura 2.5 – Cargas equivalentes de pré-esforço; (a) Cabo com uma mudança de declive abrupta; (b) Cabo parabólico. [2].

As expressões para o cálculo das cargas equivalentes ao pré-esforço em vigas simplesmente apoiadas, correspondentes aos perfis ilustrados na Figura 2.6, são apresentadas nas equações (2), (3) e (4), respetivamente.

$$W_{up} = \frac{8 \cdot P \cdot e}{L^2} \quad (2)$$

$$W_{up} = \frac{4 \cdot P \cdot e}{L} \quad (3)$$

$$W_{up} = \frac{P \cdot e}{a \cdot L} \quad (4)$$

Onde: P é a força de pré-esforço, e é a excentricidade do cabo de pré-esforço, L é comprimento do vão, a é uma percentagem e W_{up} é a força equivalente de pré-esforço.

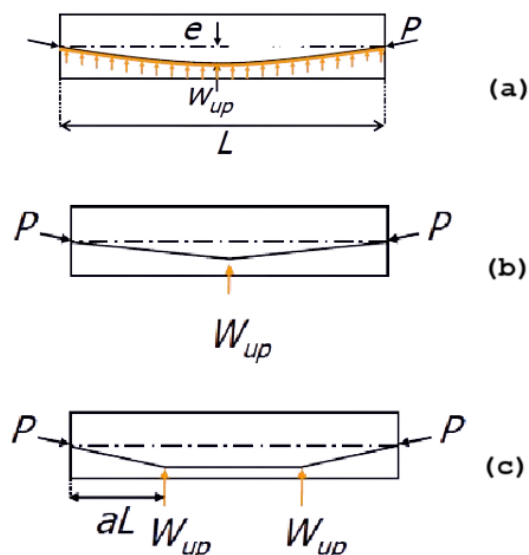


Figura 2.6 – Cargas Equivalentes de pré-esforço; (a) Carga distribuída; (b) Carga pontual à meio vão; (c) Duas cargas pontuais a distâncias de aL e $L - aL$ dos apoios, respetivamente [7] (adaptado).

2.3. Modelação de cabos de pré-esforço

No detalhamento dos cabos de pré-esforço e das armaduras, é fundamental satisfazer os pressupostos de análise, garantir a correta colocação do betão, assegurar a durabilidade, e, após os cálculos de dimensionamento, elaborar os desenhos para a construção. [7]. Os cabos de pré-esforço com perfis parabólicos incorporados em elementos de betão são principalmente utilizados para resistir a cargas distribuídas de forma uniforme ou quase uniforme, e o comportamento mecânico destes cabos pode ser facilmente estimado através do conceito de cargas equivalentes [8].

2.3.1. Geometrias práticas e equações que definem os traçados

Para uma viga simplesmente apoiada pós-tensionada, sujeita a uma carga uniformemente distribuída significativa, é adotado um traçado parabólico para os cabos [7], conforme ilustrado na Figura 2.7. O traçado é definido pela equação (5).

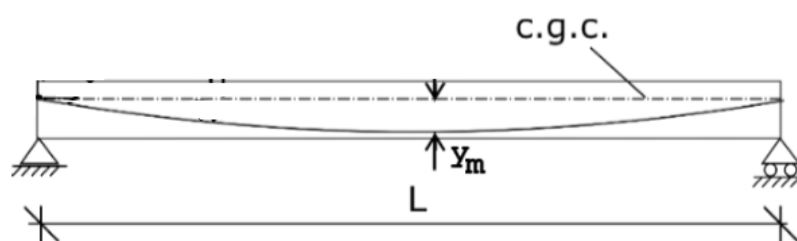


Figura 2.7 – Viga simplesmente apoiada com cabo de pré-esforço de traçado parabólico, onde y_m representa a flecha, e c.g.c. corresponde ao eixo neutro [8].

$$y(x) = \frac{4 \cdot y_m}{L^2} \cdot (L - x) \quad (5)$$

Onde: y_m é a distância entre o ponto de excentricidade inicial e o ponto de excentricidade máxima, L é o comprimento do vão, e x são os pontos ao longo do vão.

Numa viga contínua, é raro utilizar um perfil de cabo que consista numa única parábola, como ilustrado na Figura 2.7; um perfil de cabo mais adequado nestas condições é composto por uma série de segmentos parabólicos, cada um com uma forma diferente, sendo côncavos nos vãos e convexas sobre os apoios interiores, conforme ilustrado na Figura 2.8, e os segmentos convexas são necessários para evitar dobras acentuadas no cabo na zona dos apoios internos [1].



Figura 2.8 – Viga contínua com traçados parabólicos [7].

Em vigas contínuas sujeitas a múltiplos vãos e carregamentos, os segmentos parabólicos em torno de um ponto de excentricidade máxima podem não apresentar simetria [7]. Cada segmento parabólico deve satisfazer duas condições: possuir declive nulo nos pontos de excentricidade máxima e garantir que os declives em ambos os lados de um ponto de inflexão sejam idênticos [7]. Assim, a equação que define um segmento parabólico é apresentada na equação (6), sendo o respetivo perfil ilustrado na Figura 2.9.

$$y(x) = y_m \cdot \left(\frac{x}{l}\right)^2 \quad (6)$$

Onde: y_m é a distância entre o ponto de excentricidade máxima e o ponto de inflexão, l é o comprimento do segmento parabólico, e x são os pontos ao longo do segmento.

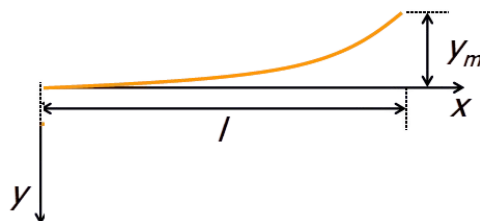


Figura 2.9 - Perfil de um segmento parabólico [7] (adaptado).

2.4. Elementos utilizados no pré-esforço

2.4.1. Betão estrutural

No âmbito da resistência à compressão em estruturas de betão, consideram-se betões com resistência característica que pode exceder 120 MPa, tratando-se de betões de resistência normal (NSC), com $f_{ck} \leq 50$ MPa, e betões de alta resistência (HSC), com $f_{ck} > 50$ MPa [9].

A resistência de referência à compressão do betão, de acordo com a norma ISO 1920-3 [10], é determinada através de ensaios em cilindros de 150×300 mm ou cubos de $150 \times 150 \times 150$ mm. No contexto da análise e dimensionamento de estruturas de betão, utiliza-se a resistência característica à compressão f_{ck} , definida pelo critério segundo o qual 5% das medições possíveis da resistência do betão especificado devem estar abaixo de f_{ck} [9]. A Tabela 2.2 apresenta uma seleção de valores característicos da resistência à compressão do betão.

Tabela 2.2 – Valores caraterísticos da resistência à compressão do betão [9] [5]. (adaptado)

Resistência à compressão do Betão													
f_{ck}	30	35	40	45	50	55	60	70	80	90	100	110	120
$f_{ck,cube}$	37	45	50	55	60	67	75	85	95	105	115	130	140
f_{cm}	38	43	48	53	58	63	68	78	88	98	108	118	128

Os principais requisitos para o betão em estruturas pré-esforçadas incluem: a utilização de betão à base de cimento Portland comum, com resistência característica geralmente igual ou superior a 50 MPa; elevada resistência inicial, essencial para a antecipação da aplicação do pré-esforço; módulo de elasticidade elevado, fundamental para a minimização das deformações do elemento estrutural; e uma composição otimizada para reduzir a fluência do betão, mitigando as perdas de pré-esforço [11].

A Figura 2.10 ilustra o impacto que a fluência tem no betão pré-esforçado.

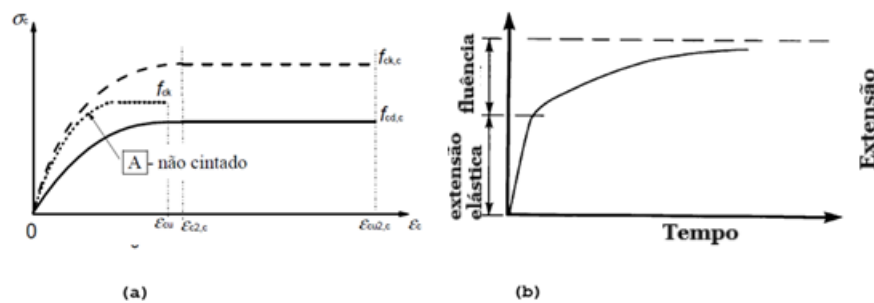


Figura 2.10 – (a) Relação tensão-extensão para o betão cintado e não cintado; (b) Influência da fluência no betão comprimido [5], [11] (adaptado).

O encurtamento do elemento pré-esforçado causado pela fluência e retração provoca um encurtamento correspondente do aço de pré-esforço que está fisicamente ligado ao betão, quer por aderência, quer por ancoragens nas extremidades do cabo, e este pode ser significativo e geralmente resulta numa perda de tensão no aço entre 150 e 300 MPa [1].

2.4.2. Aço de pré-esforço

As armaduras de pré-esforço em estruturas de betão podem ser constituídas por fios, varões e cordões, devendo apresentar baixa suscetibilidade à corrosão sob tensão, conforme estabelecido na norma EN 10138 [12] ou numa Aprovação Técnica Europeia. As propriedades das armaduras devem ser avaliadas nas condições finais da estrutura, sendo o cumprimento da EN 10138 considerado suficiente para garantir a conformidade com os requisitos da NP EN 1992-1-1. As especificações incluem a resistência à tração, a tensão limite convencional e a extensão. Adicionalmente, os produtos devem ser identificáveis, classificados quanto à relaxação e acompanhados de um certificado. As soldaduras são permitidas apenas nos fios dos cordões antes da trefilagem a frio, e as armaduras fornecidas em rolo devem respeitar a flecha máxima especificada pela EN 10138 após o desenrolamento [5].

De acordo com [7], existem diferentes formas de aço de pré-esforço, que incluem:

- Fios: uma unidade única de aço;
- Cordões: constituídos por dois, três ou sete fios enrolados para formar um cordão de pré-esforço;
- Tendão: conjunto de cordões ou fios enrolados para formar um tendão de pré-esforço;
- Cabo: um conjunto de tendões que formam um cabo de pré-esforço;
- Barras: um tendão que pode ser constituído por uma única barra de aço, com um diâmetro significativamente maior do que o de um fio.

Quanto à natureza da interface entre o betão e o aço, [7] distingue:

- Aderente: ocorre quando existe uma aderência adequada entre as armaduras de pré-esforço e o betão; as armaduras pré-tensionadas e pós-tensionadas com injeção de calda de cimento são consideradas aderentes.
- Não aderente: caracteriza-se pela ausência de aderência entre as armaduras de pré-esforço e o betão; se não for realizada a injeção de calda de cimento após a pós-tensão, as armaduras são consideradas não aderentes.

O número de cordões num cabo de pré-esforço varia de 2 a 55 cordões. Apresenta-se na Tabela 2.3, um resumo com os diâmetros e tensões de resistência à tração dos aços de pré-esforço.

Tabela 2.3 – Características dos aços de pré-esforço [1], [13]. (adaptado)

Diâmetro nominal [mm]	Designação [-]	Tensão nominal [MPa]	Módulo de elasticidade [GPa]
15.3	Y1770S7	1770	195 ± 10
15.3	Y1860S7	1860	195 ± 10
15.7	Y1770S7	1770	195 ± 10
15.7	Y1860S7	1860	195 ± 10

Apresenta-se na Figura 2.11 um diagrama típico de tensão-extensão para aços de alta resistência.

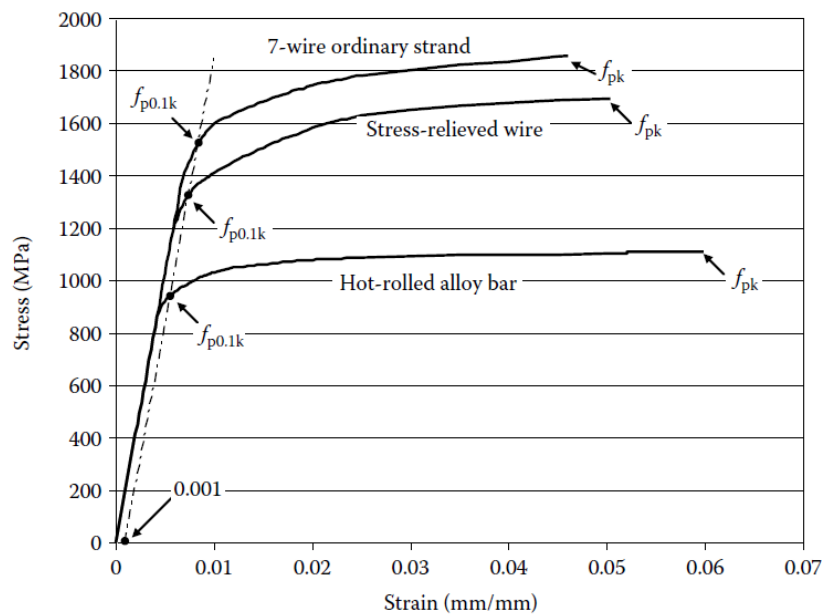


Figura 2.11 – Diagramas típicos de tensão-extensão de aços de pré-esforço [1].

2.4.3. Bainhas e ancoragens

De acordo com [9], as bainhas nos sistemas de pré-esforço têm um papel essencial na criação de cavidades para as armaduras, na definição dos seus percursos, na transferência de tensões e na proteção contra corrosão. A escolha entre bainhas metálicas onduladas, plásticas onduladas, tubos de aço lisos ou plásticos lisos deve cumprir os requisitos do projeto e ser devidamente documentada.

É fundamental que as bainhas atendam aos padrões de concepção, fabrico e instalação, garantindo a sua adequação e durabilidade no sistema estrutural [9]. Após o endurecimento do betão, as armaduras são tensionadas e fixadas em ancoragens, responsáveis pela transferência das forças de pré-esforço. Os principais tipos de ancoragens são [9]:

- Ancoragens ativas: permitem o tensionamento do cabo;

- Ancoragens passivas: não permitem o tensionamento do cabo (estão embebidas no betão);
- Dispositivos de acoplamento: são usados para ligar a extremidade de um cabo já tensionado a um segundo cabo colocado como extensão do primeiro, que será tensionado numa segunda fase.

A Figura 2.12 ilustra os três tipos básicos de ancoragens.

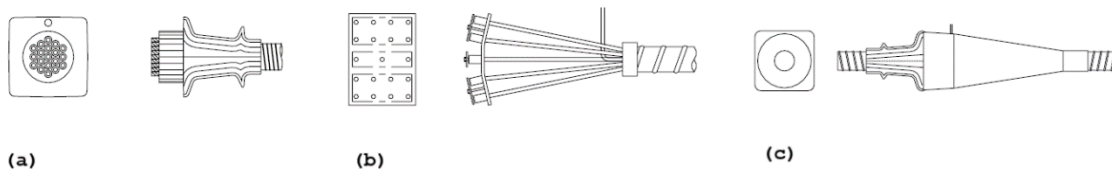


Figura 2.12 - Tipos de ancoragens. Ancoragem ativa (a). Ancoragem passiva (b). Dispositivo de acoplamento (c) [14] (adaptado).

Além disso, [7] destaca que a seleção e implementação adequadas das bainhas são fatores críticos para o desempenho e longevidade das estruturas pré-esforçadas. O controlo de qualidade e a documentação rigorosa são essenciais para assegurar a sua conformidade com os critérios funcionais e de durabilidade exigidos em aplicações de construção.

2.5. Perdas de pré-esforço

De acordo com [5], o pré-esforço aplicado ao betão deve ser considerado na análise estrutural, podendo ser tratado como uma ação ou como resistência devido à deformação e curvatura iniciais. O mesmo documento refere ainda que o pré-esforço deve ser incluído nas combinações de ações conforme a EN 1990, influenciando o momento e o esforço normal aplicados. [5] também estabelece que a contribuição das armaduras de pré-esforço para a resistência da secção é limitada à que resulta após a sua tração, considerando o efeito do pré-esforço na curva tensões-extensões.

Para evitar a rotura frágil do elemento, [5] recomenda a adoção de várias medidas, incluindo: prever armadura mínima, utilizar armaduras de pré-esforço aderentes pré-tensionadas, garantir acesso para inspeções, demonstrar a fiabilidade das armaduras, e assegurar que a fendilhação ocorre antes de se exceder a capacidade última em caso de sobrecarga ou redução do pré-esforço.

As perdas no pré-esforço são classificadas em imediatas, que ocorrem durante o tensionamento dos cabos e a transferência do pré-esforço, e em diferidas, que surgem ao longo da vida útil do elemento pré-esforçado [7]. As perdas imediatas envolvem a deformação elástica do betão, o atrito cabo-betão e a reentrada das cunhas nas ancoragens, enquanto as perdas diferidas incluem a retração e fluência do betão, além da relaxação do aço [7]. Apresenta-se na Figura 2.13 um diagrama resumindo os tipos de perdas.

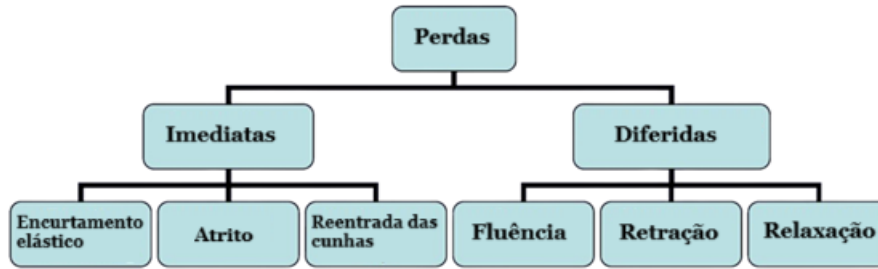


Figura 2.13 - Tipos de perdas de pré-esforço [7].

Nesta dissertação, serão abordadas exclusivamente as perdas de pré-esforço no caso da pós-tensão e para armaduras aderentes.

2.5.1. Força de tensionamento

De acordo com [5], [9], antes da transferência do pré-esforço para o betão, a tensão máxima de tração nas armaduras não deve exceder o menor dos valores indicados na equação (7). Após a transferência do pré-esforço, a força máxima de tração nas armaduras não deve ultrapassar o menor dos valores estabelecidos na equação (8).

$$\sigma_{p,max} = \min\{0,8 \cdot f_{pk}; 0,9 \cdot f_{p0,1k}\} \quad (7)$$

$$\sigma_{pm0} = \min\{0,75 \cdot f_{pk}; 0,85 \cdot f_{p0,1k}\} \quad (8)$$

Onde: $\sigma_{p,max}$ é a tensão máxima aplicada às armaduras; σ_{pm0} é a tensão nas armaduras imediatamente após a aplicação do pré-esforço; f_{pk} é o valor característico da tensão de rotura à tração do aço das armaduras de pré-esforço; $f_{p0,1k}$ é o valor característico da tensão limite convencional de proporcionalidade a 0,1 % à tração do aço das armaduras de pré-esforço.

2.6. Perdas imediatas de pré-esforço

2.6.1. Perdas devidas ao atrito

De acordo com [15], as perdas por atrito ocorrem devido às curvaturas do cabo de pré-esforço, ondulação, atrito no macaco e atrito nas ancoragens. O atrito entre o aço de pré-esforço e a bainha depende do ângulo, θ , através do qual o cabo é desviado, e do coeficiente de atrito, μ , entre a bainha e o cabo. Ainda segundo [15], devido a dificuldades na construção, é impossível manter um alinhamento completamente reto ou parabólico da bainha, o que resulta em várias ondulações ao longo do seu comprimento. Estas ondulações geram atrito adicional, expresso por um coeficiente k , cujo valor depende do tipo de armadura e do tipo de bainha ou revestimento utilizado. Segundo [5], as perdas devidas ao atrito deverão ser calculadas de acordo à equação (9).

$$\Delta P_{\mu}(x) = P_{max} \cdot (1 - e^{-\mu \cdot (\theta + k \cdot x)}) \quad (9)$$

Onde: θ é a soma dos desvios angulares ao longo do comprimento x ; μ é o coeficiente de atrito entre a armadura de pré-esforço e a bainha; k é o desvio angular parasita, por unidade de comprimento; x é a distância ao longo da armadura a partir do ponto onde a força de pré-esforço é máxima.

O valor de μ depende das características da superfície das armaduras e da bainha, da presença de ferrugem, do alongamento das armaduras e do seu traçado [5], apresentam-se na Tabela 2.4 sugestões de valores para o coeficiente de atrito. Na ausência de dados ou de uma Aprovação Técnica Europeia, os valores dos desvios angulares parasitas, k , situam-se, em geral, no intervalo $0,005 < k < 0,01$ por metro.

Tabela 2.4 - Coeficientes de atrito típicos para diferentes tipos de aços de pré-esforço e bainhas [9].

Tipo de aço de pré-esforço	Bainha metálica ondulada	Bainha de plástico ondulada	Tubo de aço liso	Tubo de plástico liso
Fios treilados a frio	0,14 – 0,18	0,08 – 0,12	0,25 – 0,30	0,08 – 0,12
Cordões	0,16 – 0,20	0,10 – 0,14	0,25 – 0,30	0,10 – 0,14
Barras nervuradas	0,63 – 0,68	-	-	-
Barras lisas	0,30 – 0,35	-	-	-

A Figura 2.14 ilustra um gráfico típico das perdas por atrito numa viga simplesmente apoiada, tensionada apenas numa extremidade e ancorada na outra.

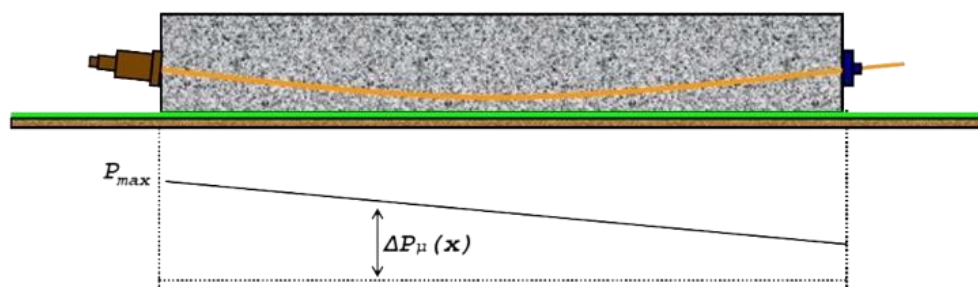


Figura 2.14 - Gráfico típico da variação do pré-esforço após as perdas por atrito numa viga simplesmente apoiada [7] (adaptado).

2.6.2. Perdas devidas à reentrada das cunhas nas ancoragens

Durante a transferência de pré-esforço do macaco para a ancoragem, ocorre algum escorregamento ou reentrada o que resulta numa perda adicional de pré-esforço, e a quantidade de escorregamento depende do tipo de ancoragem utilizada [1]. Para ancoragens do tipo cunha, utilizadas em cordões, geralmente, a reentrada atinge até 6 mm, e a perda de pré-esforço causada por esta reentrada das cunhas diminui com a distância em relação à ancoragem devido ao atrito, e pode ser desprezada na secção crítica em cabos mais longos [1]. No entanto, para cabos curtos, esta perda pode ser significativa e não deve ser ignorada no dimensionamento [1].

A Figura 2.15 ilustra um gráfico típico das perdas por reentrada das cunhas numa viga simplesmente apoiada, tensionada apenas numa extremidade e ancorada na outra.

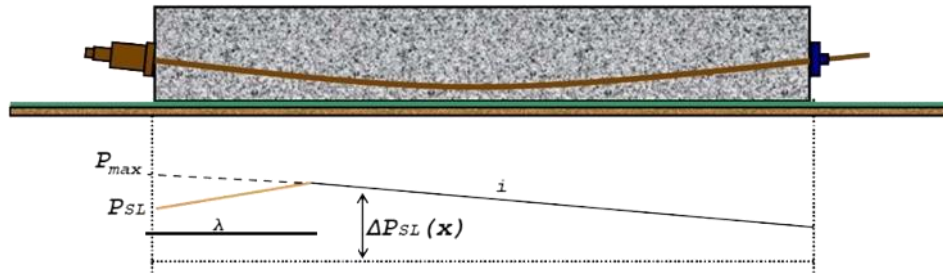


Figura 2.15 - Gráfico típico da variação do pré-esforço após as perdas por reentrada das cunhas numa viga simplesmente apoiada [7] (adaptado).

Nas equações (10) e (11), é apresentada a sequência de cálculo das perdas associadas à reentrada das cunhas, conforme indicado em [1] e em conformidade com [5]. O comprimento até onde estas perdas se manifestam é determinado pela equação (10).

$$\lambda = \sqrt{\frac{E_p \cdot A_p \cdot \Delta_{SL}}{i}} \quad (10)$$

A perda causada pela reentrada das cunhas na ancoragem é dada pela equação (11), sendo:

$$\Delta P_{SL}(x) = 2 \cdot (\lambda - x) \cdot i \quad (11)$$

Onde: E_p é o módulo de elasticidade do aço de pré-esforço; A_p é a secção transversal do aço de pré-esforço; Δ_{SL} é a reentrada das cunhas na ancoragem; i é a perda por metro linear.

2.6.3. Perdas devidas à deformação instantânea do betão

Em elementos com um único cabo, ou com múltiplos cabos tensionados simultaneamente, a deformação instantânea do betão ocorre durante o tensionamento, antes da ancoragem dos cabos. Nesse caso, as perdas por deformação elástica do betão são nulas. [1].

Num elemento com múltiplos cabos tensionados sequencialmente, o tensionamento de cada cabo gera uma perda por encurtamento elástico nos cabos previamente tensionados e ancorados. Assim, o primeiro cabo tensionado sofre a maior perda, enquanto o último não é afetado [1].

A perda de pré-esforço devido à deformação elástica do betão no momento da transferência da força é função da tensão no betão e do seu módulo de elasticidade [15]. A sequência de cálculo dessas perdas é apresentada nas equações (12) e (13), conforme [5].

$$\Delta\sigma_c(x) = \frac{\Delta P_{SL}(x)}{A} + \frac{M(x)}{I} \cdot y(x) \quad (12)$$

$$\Delta P_{el}(x) = A_p \cdot E_p \cdot \frac{\Delta\sigma_c(x)}{E_{cm}} \quad (13)$$

Onde: ΔP_{SL} é o pré-esforço após as perdas por reentrada das cunhas; A a área da secção transversal de betão; M o valor dos momentos fletor devido a força equivalente de pré-esforço e devido ao peso próprio; I é o momento de inércia da secção de betão; y é a distância do eixo neutro da viga ao centro de gravidade do cabo de pré-esforço; $\Delta\sigma_c$ é tensão no betão ao nível das armaduras, devida ao peso próprio e ao pré-esforço após as perdas por reentrada das cunhas; E_{cm} é o módulo de elasticidade secante do betão.

2.7. Perdas diferidas de pré-esforço

As perdas dependentes do tempo são calculadas considerando a redução de tensão causada pela deformação do betão devido à fluência e retração sob cargas quase-permanentes, bem como a redução de tensão nos elementos de tração devido à relaxação do material sob tensão, conforme indicado em [9]. Um método simplificado para avaliar as perdas diferidas na secção x , sob ações permanentes, é apresentado na equação (14), conforme estabelecido em [5].

$$\Delta P_{c+s+r} = A_p \cdot \frac{\varepsilon_{cs} \cdot E_p + 0.8 \cdot \Delta\sigma_{pr} + \frac{E_p}{E_{cm}} \cdot \varphi(t, t_0) \cdot \sigma_{c,QP}}{1 + \frac{E_p}{E_{cm}} \cdot \frac{A_p}{A_c} \cdot \left(1 + \frac{A_c}{I_c} \cdot z_{cp}^2\right) \cdot [1 + 0.8 \cdot \varphi(t, t_0)]} \quad (14)$$

Onde: ε_{cs} é a extensão estimada de retração, em valor absoluto; E_p é o módulo de elasticidade do aço de pré-esforço; E_{cm} é o módulo de elasticidade do betão; $\Delta\sigma_{pr}$ é a variação de tensão nas armaduras na secção x , no instante t , devida à relaxação do aço de pré-esforço, em valor absoluto; $\varphi(t, t_0)$ é o coeficiente de fluência no instante t para uma aplicação das cargas no instante t_0 ; $\sigma_{c,QP}$ é a tensão no betão ao nível das armaduras, devida ao peso próprio e ao pré-esforço inicial; A_p é a secção transversal das armaduras de pré-esforço; A_c é a área da secção transversal de betão; I_c é o momento de inércia da secção de betão; z_{cp} é a distância entre o centro de gravidade da secção de betão e as armaduras de pré-esforço.

Capítulo 3

3. Desenvolvimento do programa de cálculo automático

3.1. Requisitos e especificações do programa

3.1.1. Funcionalidades

O TendonTech é uma ferramenta desenvolvida para a modelação e cálculo automático de cabos em vigas de betão pré-esforçado. O programa considera a geometria da viga, as cargas aplicadas e as propriedades dos materiais, garantindo cálculos eficazes.

A entrada de dados realiza-se através de um ficheiro .xlsx, que inclui todos os parâmetros necessários, como geometria da viga, cargas e características dos materiais. Para a análise estrutural, o TendonTech utiliza o programa OpenSees [16], integrando a interação entre os cabos e a viga, bem como as forças resultantes do pré-esforço. Esta abordagem assegura uma simulação fiável do comportamento estrutural.

Entre as principais funcionalidades do programa destaca-se a quantificação das perdas de pré-esforço e a estimativa do pré-esforço útil nas secções críticas ao longo do tempo, permitindo uma análise detalhada do sistema. Adicionalmente, o TendonTech possibilita a exportação e visualização do traçado dos cabos em software CAD compatível.

O programa é compatível com diferentes computadores, desde que instalados o Python [17] ou o pacote Anaconda [18] e as bibliotecas necessárias, garantindo acessibilidade em diversos ambientes de trabalho.

3.1.2. Limitações

Atualmente, o TendonTech possui uma interface de utilizador baseada exclusivamente na consola, o que pode representar um desafio para utilizadores não familiarizados com a linha de comando.

A modelação geométrica dos cabos está implementada de forma *built-in*, limitando a personalização das suas posições e curvaturas ao longo da viga, o que restringe a flexibilidade do programa.

O funcionamento do TendonTech depende de bibliotecas externas, como o OpenSeesPy. Assim, é necessário ter o Python ou o pacote Anaconda instalados, bem como garantir a configuração adequada das dependências. O desempenho do programa pode ser afetado por máquinas com

configurações mais modestas ou pela complexidade dos cálculos, especialmente em modelos de maior porte, resultando em tempos de processamento variáveis.

O programa suporta vigas de um único vão ou contínuas com até quatro vãos, sem considerar consolas ou encastramentos. A visualização dos cabos é restrita ao perfil, permitindo exibir apenas um cabo por vez. A representação espacial e a visualização integrada estão limitadas a vigas com até dois vãos.

Quanto às secções transversais, o programa está limitado a vigas com secções retangulares, em “I” ou em “T”. Ainda não é possível definir secções aleatórias ou com inércia variável. No que se refere ao traçado dos cabos, este está limitado ao perfil, assumindo troços retos em planta. Para múltiplos cabos definidos, é assumido que estes são paralelos, com um máximo de 10 cabos; esta versão também considera os tensionamentos de forma sequencial, não sendo ainda possível alternar a ordem de tensionamento dos cabos.

O utilizador deve ter conhecimento prévio sobre o conceito de pré-esforço para estimar o número de cabos necessários, uma vez que o programa não realiza este cálculo automaticamente, exigindo uma definição prévia dos cabos a serem utilizados.

3.2. Interação utilizador - programa

3.2.1. Interface e fluxo de trabalho

O TendonTech é iniciado através da consola, sendo essencial que a máquina do utilizador cumpra os requisitos mínimos para o funcionamento do programa. Recomenda-se, para evitar conflitos, a criação de um ambiente virtual para isolar as dependências e garantir o adequado funcionamento do programa; a criação deste ambiente pode ser feita como indica a Tabela 3.1.

Tabela 3.1 – Criação de um ambiente virtual para isolamento das dependências externas através da linha de comandos do Windows.

```
python -m venv venv
source venv/bin/activate # No Windows use: venv\Scripts\activate
pip install -r requirements.txt # Para instalar dependências do arquivo requirements.txt, como
indicado no manual de instruções
```

Ao iniciar o programa, são apresentados três prints básicos: o nome do programa, a mensagem “Projeto de aplicação de pré-esforço” e a solução geral em função do diagrama de Magnel, como mostra a Figura 3.1.

```

TendonTech 1.0 2024:
Projeto de aplicação de pré-esforço

Solução (Magnel):
Nº Cabos = 2 cabos
Nº Cordões por cabo = 22 cordões
Ap = 33.0 Cm^2
Pmax = 4910.4 kN

```

Figura 3.1 – Interação utilizador – programa.

A interação é exclusivamente baseada em texto, com outputs gráficos (*plots*) apresentados ao longo do cálculo. Não há planos, nesta versão, para evoluir para uma interface gráfica ou outro tipo de interação.

3.2.2. Entrada de dados

Os dados necessários para o cálculo são fornecidos através de um ficheiro Excel denominado TT_InputData.xlsx, que contém parâmetros pré-preenchidos. Os dados genéricos que não influenciam significativamente os resultados são fixos, enquanto os dados editáveis estão destacados em azul para facilitar a identificação, como ilustrado na Figura 3.2.

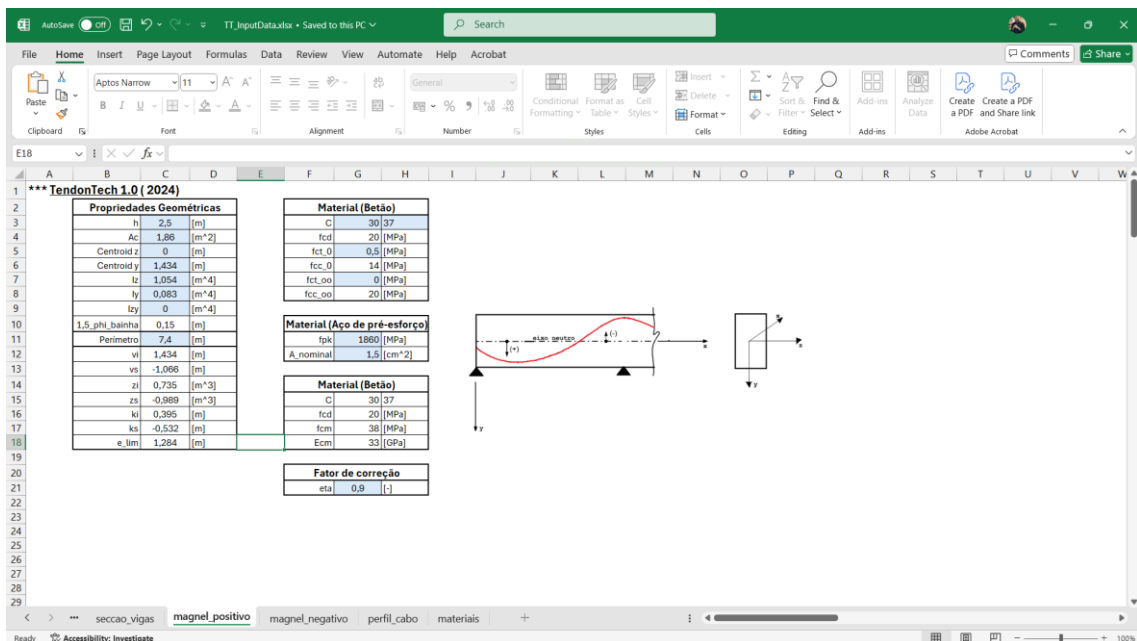


Figura 3.2 – Entrada de dados.

Não há validação automática para todos os dados inseridos. Caso sejam introduzidos valores incorretos ou estejam ausentes dados obrigatórios, o programa reporta o erro e a sua localização após a execução. Cabe ao utilizador corrigir os dados no ficheiro e reiniciar o programa.

Os parâmetros obrigatórios para o cálculo estão destacados a azul no ficheiro de entrada. Se não forem preenchidos, o programa pode assumir valores genéricos para alguns casos ou reportar erros para outros.

3.2.3. Interação utilizador - programa

Após a leitura dos dados, o programa processa os cálculos e apresenta os resultados de forma incremental. À medida que avança, são gerados prints dos resultados na consola e diagramas gráficos para auxiliar a visualização.

Os resultados são apresentados em três formatos principais:

- Consola: resultados organizados em tabelas.
- Ficheiros Excel (.xlsx): contêm valores específicos para análise posterior.
- Ficheiro DXF: permite a visualização gráfica e à escala real do perfil dos cabos em software CAD.
- Relatório final (.docx e .pdf): inclui todos os dados e resultados detalhadamente discriminados.

3.2.4. Mensagens e tratamento de erros

Os erros mais comuns referem-se a problemas na entrada de dados. Nestes casos, o programa fornece mensagens indicando a natureza do erro e possíveis correções, como indicado na Figura 3.3.

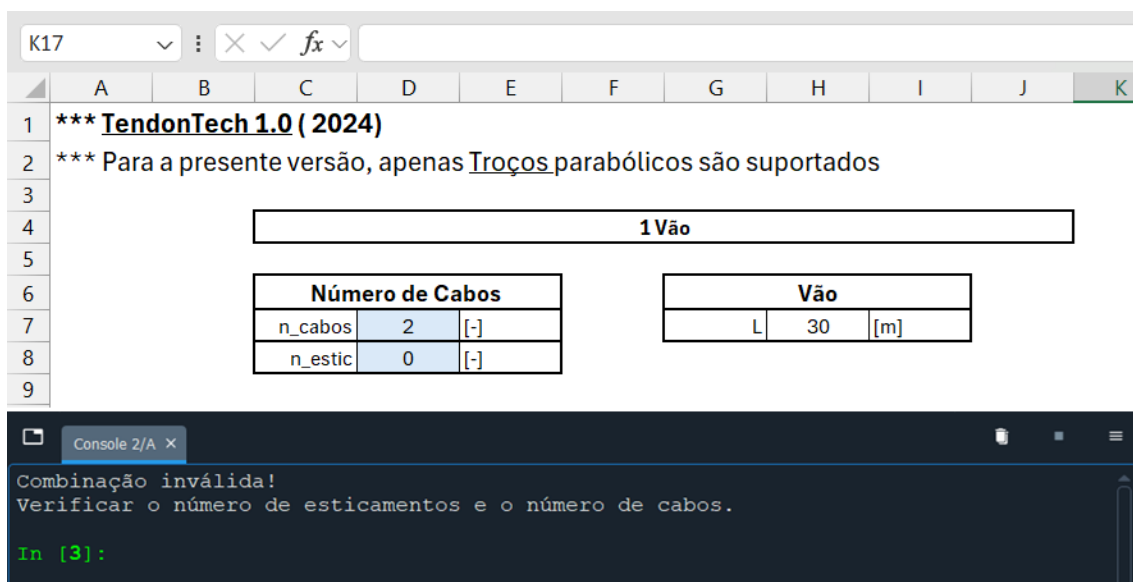


Figura 3.3 - Erro comum à entrada dos dados.

Para erros técnicos mais complexos, como falhas em dependências externas, o programa reporta o erro, mas a responsabilidade de interpretação e resolução recai sobre o utilizador, como exemplificado na Figura 3.4.

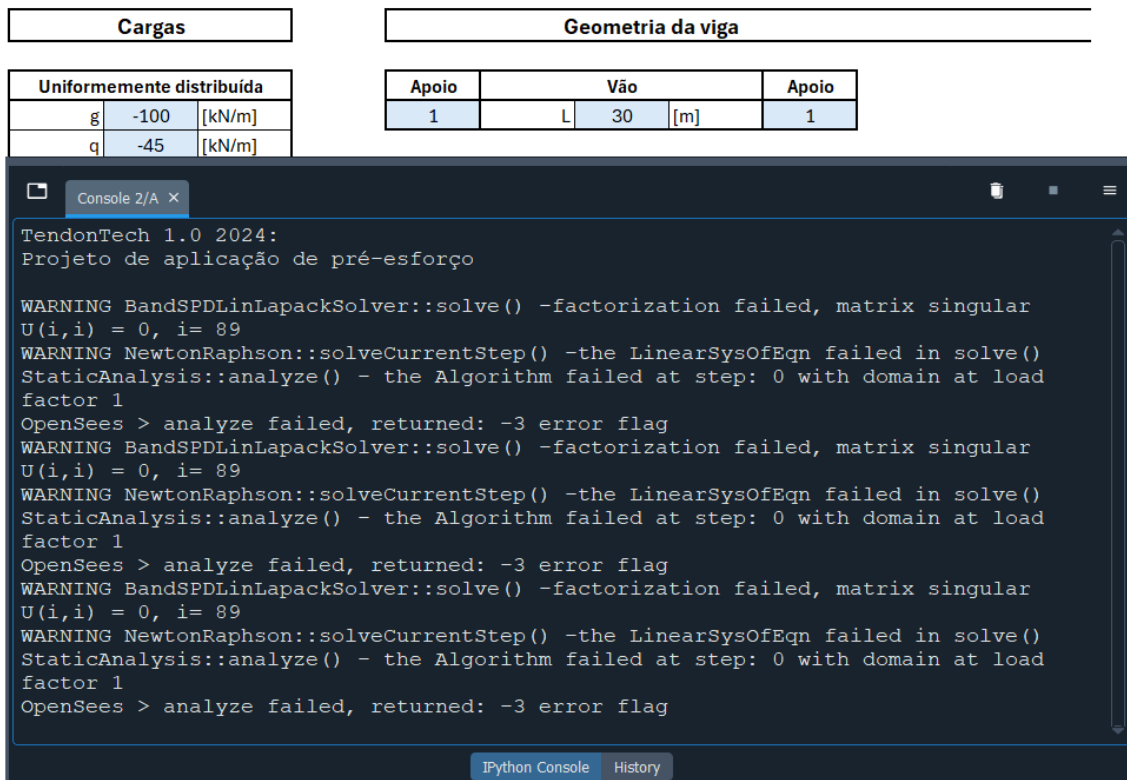


Figura 3.4 - Erro complexo. Dependência externa OpenSeesPy.

O erro apresentado na Figura 3.4 resulta da identificação de um sistema estrutural hipostático. Este erro é automaticamente detetado e reportado pelo programa por meio da interface com o OpenSeesPy, considerando as seguintes condições: “1” representa um apoio simples, “2” um apoio duplo e “3” um encastramento.

3.2.5. Documentação e suporte

O TendonTech é acompanhado de um manual do utilizador que inclui instruções detalhadas e um exemplo prático de aplicação passo a passo. O programa não possui comandos adicionais de ajuda ou suporte interno para questões técnicas.

3.3. Liguagem de programação: Python

Python é uma linguagem de programação poderosa e de fácil aprendizagem, caracterizada pela sua sintaxe elegante e tipagem dinâmica; a sua abordagem à programação orientada a objetos, aliada à natureza interpretada, torna-a ideal para *scripting* e desenvolvimento rápido de aplicações em diversas plataformas [19].

O interpretador Python, juntamente com a sua extensa biblioteca padrão, é *opensource* e está disponível gratuitamente para todas as principais plataformas no site oficial [17], permitindo distribuição livre.

3.4. Implementação do programa

Nesta secção descreve-se o processo técnico adotado para desenvolver o programa TendonTech, com foco na análise estrutural, modelação geométrica e quantificação das perdas de pré-esforço; este processo envolveu a utilização de ferramentas e bibliotecas específicas para garantir a precisão dos cálculos e a eficiência na integração com softwares CAD.

A implementação foi realizada em Python, escolhida pela sua versatilidade e robustez para tarefas de engenharia e modelação; bibliotecas como OpenSeesPy foram fundamentais para a análise estrutural, enquanto outras ferramentas, como NumPy, Matplotlib e bibliotecas de CAD, como o ezdxf, foram integradas para auxiliar na manipulação de dados, visualização gráfica e exportação para softwares de desenho técnico.

3.4.1. Definição geométrica da viga

A definição geométrica da viga é configurada conforme os critérios do utilizador. Na versão atual do TendonTech, a aplicação está limitada a secções transversais retangulares, em “T” ou em “I”, assumindo inércia constante ao longo do comprimento da viga. Consideremos, como exemplo, uma viga contínua de três vãos, conforme ilustrado na Figura 3.5.

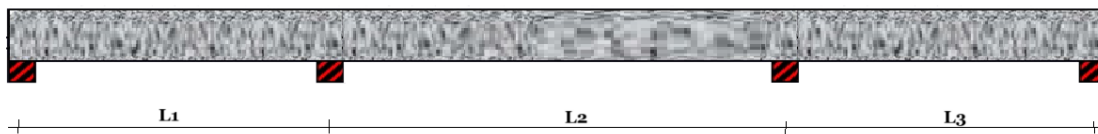


Figura 3.5 - Viga contínua com 3 tramos.

Para a viga representada na Figura 3.5, a secção em “T” revela-se a escolha mais apropriada devido à sua eficiência estrutural, permitindo concentrar material nas regiões críticas onde é mais eficaz. Como ponto de partida, pode-se estimar a altura da viga utilizando a regra prática $L/15$ a $L/20$. Considerando um vão $L = 35$ m, obtém-se uma altura estimada de $h = 1,75$ m. A Figura 3.6 apresenta uma representação da secção, enquanto a Tabela 3.2 fornece as propriedades geométricas detalhadas da secção adotada.

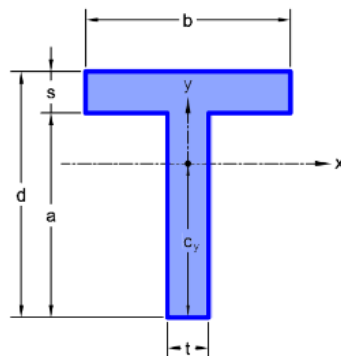


Figura 3.6 - Secção transversal em “T”.

Onde:

- $b = 1,30$ m
- $s = 0,60$ m
- $a = 1,15$ m
- $t = 0,60$ m
- $d = 1,75$ m

Estes valores originam as propriedades geométricas detalhadas na Tabela 3.2.

Tabela 3.2 - Propriedades geométricas da secção transversal em “T”.

Propriedades geométricas da secção transversal	
Área, A_c [m ²]	1,470
Momento de inércia, I_z [m ⁴]	0,380
Altura, h [m]	1,750
Perímetro, u [m]	6,100
Distância do eixo neutro à fibra inferior, v_i [m]	1,039
Distância do eixo neutro à fibra superior, v_s [m]	-0,711
Módulo de Resistência da fibra inferior, z_i [m ³]	0,365
Módulo de Resistência da fibra superior, z_s [m ³]	-0,534
Distância do centro geométrico ao limite superior do Núcleo Central, k_i [m]	0,249
Distância do centro geométrico ao limite inferior do Núcleo Central, k_s [m]	-0,363

Nota: A convenção de sinais utilizada para os cálculos segue a orientação apresentada na Figura 3.7.

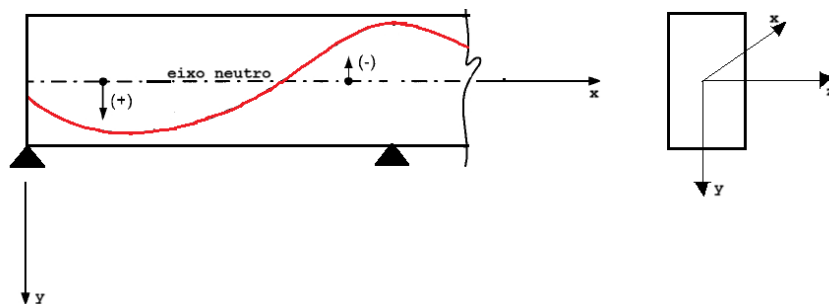


Figura 3.7 - Convenção de sinais.

A Tabela 3.3 apresenta o bloco de código necessário para importar a geometria da viga a partir da base de dados TT_InputData.xlsx.

Tabela 3.3 – Código para importação de dados de entrada do TendonTech: Geometria.

```
from openpyxl import load_workbook
wb = load_workbook(r"C:/Users/luton/11. TendonTech 1.0 (BETA)/TT_InputData.xlsx",
data_only=True)
magnel_positivo = wb['magnel_positivo']
def dados():
    global Ac, Iz, h, u, ,vi, vs, zi, zs, ki, ks, \
    Ac = (magnel_positivo['C4'].value) # [m^2]
    Iz = (magnel_positivo['C7'].value) # [m^4]
    h = (magnel_positivo['C3'].value) #[m]
    u = (magnel_positivo['C11'].value) #[m]
    vi = (magnel_positivo['C12'].value) #[m]
    vs = (magnel_positivo['C13'].value) #[m]
    zi = (magnel_positivo['C14'].value) #[m^3]
    zs = (magnel_positivo['C15'].value) #[m^3]
    ki = (magnel_positivo['C16'].value) #[m]
    ks = (magnel_positivo['C17'].value) #[m]
dados()
```

3.4.2. Análise estrutural com OpenSeesPy: Análise estática, Diagrama de Magnel e Força de tensionamento

O OpenSees (Open System for Earthquake Engineering Simulation) é um *framework* (conjunto de bibliotecas de software) orientado a objetos e de código aberto, desenvolvido para a modelação e análise de sistemas estruturais e geotécnicos através do método dos elementos finitos. Suporta execuções em modo sequencial e paralelo, permitindo a simulação do comportamento dessas estruturas quando submetidas a sismos e outras ações [20], [21]. Escrito principalmente em C++, o OpenSees integra bibliotecas numéricas em Fortran e C para a resolução de equações lineares, bem como para rotinas de materiais e elementos [20].

O objetivo principal do desenvolvimento do OpenSees é melhorar a modelação e a simulação computacional na engenharia sísmica por meio de desenvolvimento de código aberto, educação e formação e discussão e partilha pela comunidade científica [20], [21]. Adicionalmente, o OpenSees disponibiliza diversas aplicações no seu site oficial, promovendo a utilização e a expansão do conjunto de bibliotecas para novas abordagens em engenharia sísmica [21].

O OpenSeesPy é a versão do OpenSees integrada ao interpretador Python, baseada no mesmo código-fonte. Mantém total equivalência em recursos e funcionalidades, oferecendo as mesmas capacidades de modelação e análise [22].

Na sequência ao exemplo ilustrado na Figura 3.5, já com a secção transversal definida, pode estimar-se o peso próprio da viga:

$$g = A_c \cdot \gamma_c = 1,470 \cdot 25 = 38 + R_{cp} = 50 \text{ kN/m}$$

Onde: A_c representa a área da secção transversal do betão; γ_c o peso volúmico do betão armado; e R_{cp} corresponde a uma estimativa das cargas permanentes adicionais.

Para a sobrecarga, considera-se 50% do peso próprio, com $q = 25 \text{ kN/m}$. Assumem-se os seguintes comprimentos: $L_1 = 28 \text{ m}$, $L_2 = 35 \text{ m}$ e $L_3 = 28 \text{ m}$. Na Tabela 3.4, apresenta-se o bloco de código responsável pela importação dos dados relevantes para a análise estática da viga.

Tabela 3.4 – Código para importação de dados de entrada do TendonTech: Análise estática

```

from openpyxl import load_workbook
from openpyxl.chart import ScatterChart, Reference, Series
import openseespy.opensees as ops
import matplotlib.pyplot as plt
import pandas as pd
import opsvis as opsv
import sympy as sp
import numpy as np
from PIL import Image
import sys

wb = load_workbook(r"C:/Users/luton/11. TendonTech 1.0 (BETA)/TT_InputData.xlsx",
data_only=True)
cargas_vigas = wb['cargas_vigas']
n_vaos = int(cargas_vigas['G4'].value) # número de vão pretendidos, [-]
print('\nTendonTech 1.0 2024:\nProjeto de aplicação de pré-esforço\n')
if n_vaos <= 0 or not n_vaos or n_vaos > 4:
    print('\nNa presente versão, apenas são suportadas vigas contínuas com até 4
vãos.\nTendonTech 1.0 2024')
    sys.exit()
elif n_vaos == 3:
    cargas_vigas = wb['cargas_vigas']
    magnel_positivo = wb['magnel_positivo']
    magnel_negativo = wb['magnel_negativo']
    perfil_cabo = wb['perfil_cabo']
    materiais = wb['materiais']
    def dados():
        global L1, L2, L3, L, H1, H2, H3, H4, H5, H6, H7, H8, H9, H10, \
            fck, fck_, fcm, Ecm, \

```

```

Ac, Iz, \
n_cabos, n_estic, g, q, psi_2, \
apoio_1, apoio_2, apoio_3, apoio_4, simples, duplo, fixo, apoios
L1 = (cargas_vigas['I16'].value) # [m], vão 1
L2 = (cargas_vigas['M16'].value) # [m], vão 2
L3 = (cargas_vigas['Q16'].value) # [m], vão 3
L = float(L1 + L2 + L3) # [m], comprimento total
H1 = 0.4 * L1 # [m], troço 1
H2 = 0.5 * L1 # [m], troço 2
H3 = 0.1 * L1 # [m], troço 3
H4 = 0.2 * L2 # [m], troço 4
H5 = 0.3 * L2 # [m], troço 5
H6 = 0.4 * L2 # [m], troço 6
H7 = 0.1 * L2 # [m], troço 7
H8 = 0.2 * L3 # [m], troço 8
H9 = 0.3 * L3 # [m], troço 9
H10 = 0.5 * L3 # [m], troço 10
fck = (materiais['D6'].value) #[MPa]
fcm = (materiais['D12'].value) # [MPa]
Ecm = (materiais['D13'].value) * 10**6 # [kN/m2]
Ac = (magnel_positivo['C4'].value) # [m^2]
Iz = (magnel_positivo['C7'].value) # [m^4]
h = (magnel_positivo['C3'].value) #[m]
g = (cargas_vigas['D10'].value) #kN/m
q = (cargas_vigas['D11'].value) #kN/m
psi_2 = (cargas_vigas['D12'].value) #[-]
apoio_1 = (cargas_vigas['G16'].value)
apoio_2 = (cargas_vigas['K16'].value)
apoio_3 = (cargas_vigas['O16'].value)
apoio_4 = (cargas_vigas['S16'].value)
simples = [0, 1, 0] #apoio simples
duplo = [1, 1, 0] #apoio duplo
fixo = [1, 1, 1] #apoio encastrado
apoios = {
    1: simples,
    2: duplo,
    3: fixo
}
dados()

```

Para uma quantificação eficaz das perdas de pré-esforço, recomenda-se a determinação dos esforços na viga em intervalos mínimos de 1 metro. Na Tabela 3.5, apresenta-se o bloco de código que realiza a divisão adequada do comprimento da viga para este propósito.

Tabela 3.5 – Código para a divisão do comprimento total da viga em intervalos de até 1 metro.

```
def dividir_comprimento(L1, L2, L3):
    num_intervalos = int(L)
    num_nos = num_intervalos + 1
    intervalos = list(range(1, num_intervalos + 1))
    nos = [i for i in range(num_intervalos)]
    if L % 1 != 0:
        nos.append(round(L, 3))
    else:
        nos.append(int(L))
    nos_id = list(range(num_nos))
    pontos_importantes = [H1, H1 + H2, L1, \
                          L1 + H4, L1 + H4 + H5, L1 + H4 + H5 + H6, L1 + L2, \
                          L1 + L2 + H8, L1 + L2 + H8 + H9]
    for ponto in pontos_importantes:
        if ponto not in nos:
            nos.append(round(ponto, 3))
            nos.sort()
    num_intervalos = (len(nos)) - 1
    intervalos = list(range(1, num_intervalos + 1))
    num_nos = (len(nos))
    nos_id = list(range(num_nos))
    indice_H2 = nos.index(round(H1 + H2, 3))
    indice_L1 = nos.index(round(L1, 3))
    indice_H4 = nos.index(round(L1 + H4, 3))
    indice_H6 = nos.index(round(L1 + H4 + H5 + H6, 3))
    indice_L2 = nos.index(round(L1 + L2, 3))
    indice_H8 = nos.index(round(L1 + L2 + H8, 3))
    return num_intervalos, num_nos, intervalos, nos, nos_id, \
           indice_H2, indice_L1, indice_H4, indice_H6, indice_L2, indice_H8
num_intervalos, num_nos, intervalos, nos, nos_id, \
indice_H2, indice_L1, indice_H4, indice_H6, indice_L2, indice_H8 =
dividir_comprimento(L1, L2, L3)
```

Nesta versão do TendonTech, o processo construtivo adotado é o de “cimbra ao solo”. Para uma viga contínua com três vãos, considera-se que a betonagem será realizada em três fases, conforme

ilustrado na Figura 3.8. Na Tabela 3.6, apresenta-se o bloco de código necessário para simular este processo construtivo.

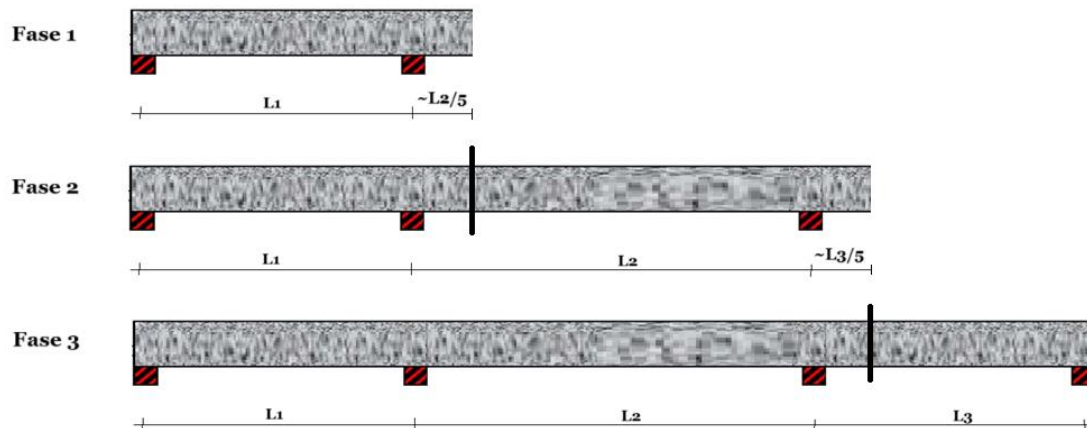


Figura 3.8 - Processo construtivo: Cimbra ao solo.

Tabela 3.6 – Código para simulação do processo construtivo: Cimbra ao solo

```
def cimbre_ao_solo():
    def configurar_modelo(pontos, numero_intervalos, verificar_apoio_3,
verificar_apoio_4):
        ops.wipe()
        ops.model('basic', '-ndm', 2, '-ndf', 3)
        ops.section('Elastic', 1, Ecm, Ac, Iz)
        for i, x in enumerate(pontos, start=1):
            ops.node(i, x, 0)
        ops.geomTransf('Linear', 1)
        for i in range(1, numero_intervalos + 1):
            ops.element('elasticBeamColumn', i, i, i + 1, Ac, Ecm, Iz, 1)
        if any(x is None or x <= 0 for x in [L1, L2, L3]):
            print("Atenção, nenhum comprimento pode ser negativo, nulo ou None.")
            sys.exit()
        if apoio_1 in apoios:
            ops.fix(nos_id[0] + 1, *apoios[apoio_1])
        else:
            print("Atenção, verificar as condições no Apoio da extremidade esquerda.")
            sys.exit()
        if apoio_2 in apoios:
            ops.fix(indice_L1 + 1, *apoios[apoio_2])
        else:
            print("Atenção, verificar as condições no Apoio interior esquerdo.")
            sys.exit()
```

```

if verificar_apoio_3:
    if apoio_3 in apoios:
        ops.fix(indice_L2 + 1, *apoios[apoio_3])
    else:
        print("Atenção, verificar as condições no Apoio interior direito.")
        sys.exit()
if verificar_apoio_4:
    if apoio_4 in apoios:
        ops.fix(nos_id[-1] + 1, *apoios[apoio_4])
    else:
        print("Atenção, verificar as condições no Apoio da extremidade direita!")
        sys.exit()
def definir_carregamento(inicio, numero_intervalos, carga):
    ops.timeSeries('Constant', 1)
    ops.pattern('Plain', 1, 1)
    ops.eleLoad('-ele', *range(inicio, numero_intervalos + 1), '-type', '-beamUniform',
carga)
def realizar_analise():
    ops.system('BandSPD')
    ops.numberer('RCM')
    ops.constraints('Plain')
    ops.integrator('LoadControl', 1.0)
    ops.test('NormDispIncr', 1.0e-12, 10, 3)
    ops.algorithm('Newton')
    ops.analysis('Static')
    ops.initialize()
    ok = ops.analyze(1)
def salvar_resultados(displ_file, nmv_file, excel_file):
    ops.printModel('-JSON', '-file', displ_file, '-node')
    ops.printModel('-JSON', '-file', nmv_file, '-ele')
    with open(displ_file, 'r') as file:
        node_disp_data = [line.strip().split() for line in file]
    with open(nmv_file, 'r') as file:
        ele_nmv_data = [line.strip().split() for line in file]
    node_disp_df = pd.DataFrame(node_disp_data)
    ele_nmv_df = pd.DataFrame(ele_nmv_data)
    with pd.ExcelWriter(excel_file) as writer:
        node_disp_df.to_excel(writer, sheet_name='Disp', index=False, header=False)
        ele_nmv_df.to_excel(writer, sheet_name='NMV', index=False, header=False)
def plotar_diagrama_momento(titulo, sfacM=5.e-4):
    opsv.section_force_diagram_2d('M', sfacM)

```

```

plt.title(titulo)
plt.show()
def plotar_modelo_estatico(titulo):
    opsv.plot_model(node_labels=1, element_labels=1)
    plt.title(titulo)
    plt.show()
def plotar_cargas(titulo):
    opsv.plot_loads_2d(nep=3, sfac=True, fig_wi_he=False, fig_lbrt=False,
fmt_model_loads={'color': 'black', 'linestyle': 'solid', 'linewidth': 1.2, 'marker': '', 'markersize':
1}, node_supports=True, truss_node_offset=0, ax=False)
    plt.title(titulo)
    plt.show()
def extrair_valores_momento(excel_file, sheet_name):
    wb = load_workbook(excel_file, data_only=True)
    CP_NMV = wb[sheet_name]
    valores_coluna_I = []
    valor_inicial = CP_NMV['I7'].value
    if valor_inicial is not None:
        valores_coluna_I.append(valor_inicial)
    indice = 8
    while True:
        celula = f'I{indice}'
        valor = CP_NMV[celula].value
        if valor is None:
            break
        valores_coluna_I.append(valor)
        indice += 8
    return valores_coluna_I
def fase_1():
    configurar_modelo(nos[:indice_H4 + 1], len(nos[:indice_H4 + 1]) - 1,
verificar_apoio_3=False, verificar_apoio_4=False)
    definir_carregamento(1, len(nos[:indice_H4 + 1]) - 1, g)
    realizar_analise()
    salvar_resultados('fase_1_Displ.txt', 'fase_1_NMV.txt', 'fase_1.xlsx')
    plotar_diagrama_momento('Diagrama de momento fletor (fase 1)')
    plotar_modelo_estatico('Modelo estático da viga (fase 1)')
    plotar_cargas('Cargas (fase 1)')
    return extrair_valores_momento('fase_1.xlsx', 'NMV')
def fase_2():
    configurar_modelo(nos[:indice_H8 + 1], len(nos[:indice_H8 + 1]) - 1,
verificar_apoio_3=True, verificar_apoio_4=False)

```

```

def definir_carregamento(len(nos[:indice_H4 + 1]), len(nos[:indice_H8 + 1]) - 1, g)
realizar_analise()
salvar_resultados('fase_2_Displ.txt', 'fase_2_NMV.txt', 'fase_2.xlsx')
plotar_diagrama_momento('Diagrama de momento fletor (fase 2)')
plotar_modelo_estatico('Modelo estático da viga (fase 2)')
plotar_cargas('Cargas (fase 2)')
return extrair_valores_momento('fase_2.xlsx', 'NMV')
def fase_3():
    configurar_modelo(nos, num_intervalos, verificar_apoio_3=True,
verificar_apoio_4=True)
    definir_carregamento(len(nos[:indice_H8 + 1]), num_intervalos, g)
    realizar_analise()
    salvar_resultados('fase_3_Displ.txt', 'fase_3_NMV.txt', 'fase_3.xlsx')
    plotar_diagrama_momento('Diagrama de momento fletor (fase 3)')
    plotar_modelo_estatico('Modelo estático da viga (fase 3)')
    plotar_cargas('Cargas (fase 3)')
    return extrair_valores_momento('fase_3.xlsx', 'NMV')
M_fase_1_string = fase_1()
M_fase_2_string = fase_2()
M_fase_3_string = fase_3()
M_fase_1 = [float(value) for value in M_fase_1_string]
M_fase_2 = [float(value) for value in M_fase_2_string]
M_fase_3 = [float(value) for value in M_fase_3_string]
return M_fase_1, M_fase_2, M_fase_3
M_fase_1, M_fase_2, M_fase_3 = cimbra_ao_solo()
def soma_momentos_cimbra_ao_solo(arr1, arr2, arr3):
    max_comprimento = max(len(arr1), len(arr2), len(arr3))
    arr1 = arr1 + [0] * (max_comprimento - len(arr1))
    arr2 = arr2 + [0] * (max_comprimento - len(arr2))
    arr3 = arr3 + [0] * (max_comprimento - len(arr3))
    M_fases = [a + b + c for a, b, c in zip(arr1, arr2, arr3)]
    return M_fases
M_fases = soma_momentos_cimbra_ao_solo(M_fase_1, M_fase_2, M_fase_3)

```

Os gráficos das situações de carga e os respetivos diagramas de momentos fletores, gerados a partir do código apresentado na Tabela 3.6, estão ilustrados na Figura 3.9, Figura 3.10 e Figura 3.11.

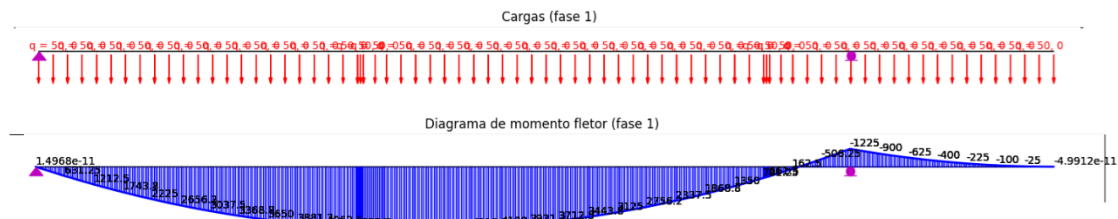


Figura 3.9 - Cimbra ao solo, fase 1: diagrama de momentos fletores.

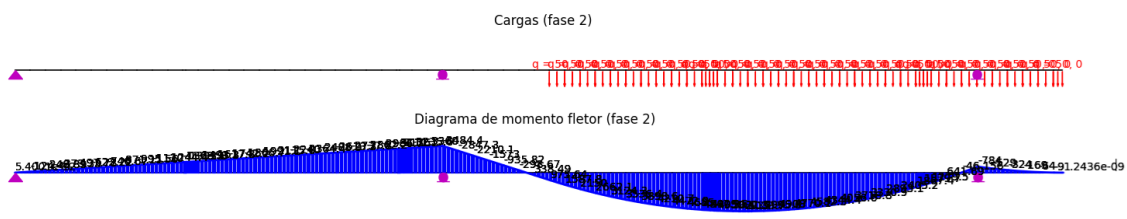


Figura 3.10 – Cimbra ao solo, fase 2: diagrama de momentos fletores.

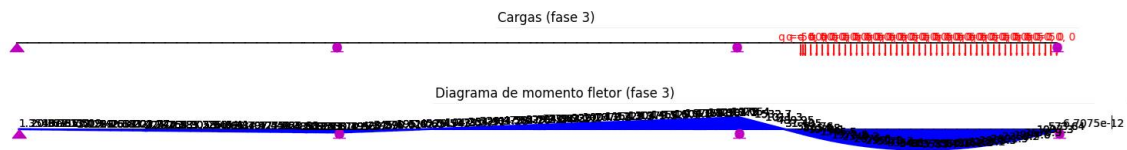


Figura 3.11 - Cimbra ao solo, fase 3: diagrama de momentos fletores.

Com os esforços internos na viga devido à carga permanente (g) já calculados, procede-se às combinações de ações para determinar os momentos fletores nas secções críticas. Em seguida, utiliza-se o diagrama de Magnel para estimar o pré-esforço útil nessas secções, considerando exclusivamente as cargas de serviço.

A interpretação geométrica das condições de desigualdade de tensões foi inicialmente explorada por Magnel [3]. As tensões admissíveis no betão e no aço são geralmente definidas pelos regulamentos; a seguinte notação será utilizada para a descrição das tensões no betão:

- f_{ct}^0 : valor admissível inicial da tensão de tração no betão
- f_{cc}^0 : valor admissível inicial da tensão de compressão no betão
- f_{cc}^∞ : valor admissível a longo prazo da tensão de compressão no betão
- f_{ct}^∞ : valor admissível a longo prazo da tensão de tração no betão

A equação (1) pode ser reestruturada de diferentes formas, de modo a destacar uma variável específica [3]. Apresenta-se na Figura 3.12, as bases para as “Inequações de Magnel”.

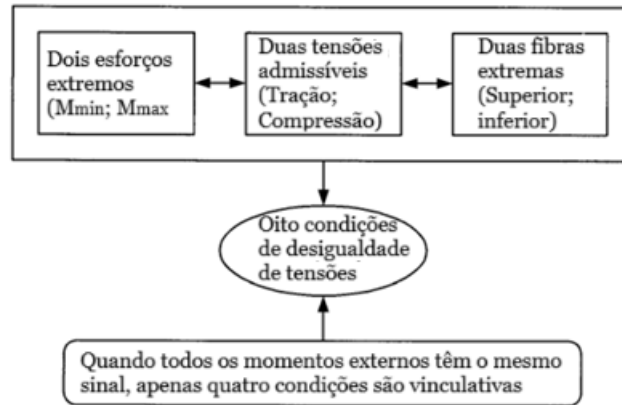


Figura 3.12 - Bases para as condições de tensão [3].

“Como dois estados de carregamento extremos são geralmente críticos e, para cada um, devem ser especificadas duas tensões admissíveis, é necessário considerar pelo menos quatro tensões admissíveis na análise. Sob carregamento de flexão, as tensões máximas ocorrem nas fibras extremas, permitindo derivar oito inequações que comparam as tensões em serviço com as admissíveis”[3].

Uma das formas práticas de escrever estas condições, é em função da excentricidade do cabo (e), como indicam as equações (15), (16), (17), (18) e (19) na Tabela 3.7:

Tabela 3.7 - Inequações de Magnel.

I	$e \leq -k_s + \frac{1}{P_\infty} \cdot \eta \cdot (M_{min} - f_{ct}^0 \cdot z_s)$	(15)
II	$e \leq -k_i + \frac{1}{P_\infty} \cdot \eta \cdot (M_{min} + f_{cc}^0 \cdot z_i)$	(16)
III	$e \geq -k_s + \frac{1}{P_\infty} \cdot (M_{max}^{Raro} + f_{cc}^\infty \cdot z_s)$	(17)
IV	$e \geq -k_i + \frac{1}{P_\infty} \cdot (M_{max}^{QP} - f_{ct}^\infty \cdot z_i)$	(18)
V	$e \leq e_{lim}$	(19)

Onde: P_∞ representa o pré-esforço útil; M_{min} corresponde ao valor do momento mínimo devido às ações permanentes; M_{max}^{Raro} refere-se ao valor do momento máximo resultante da combinação rara de ações; M_{max}^{QP} é o momento máximo associado à combinação quase-permanente de ações; $\eta \approx P_\infty/P_{max}$; e_{lim} é a excentricidade limite do cabo, tendo em conta os limites da secção de betão.

A partir das inequações, é possível obter o diagrama de Magnel, conforme ilustrado na Figura 3.13.

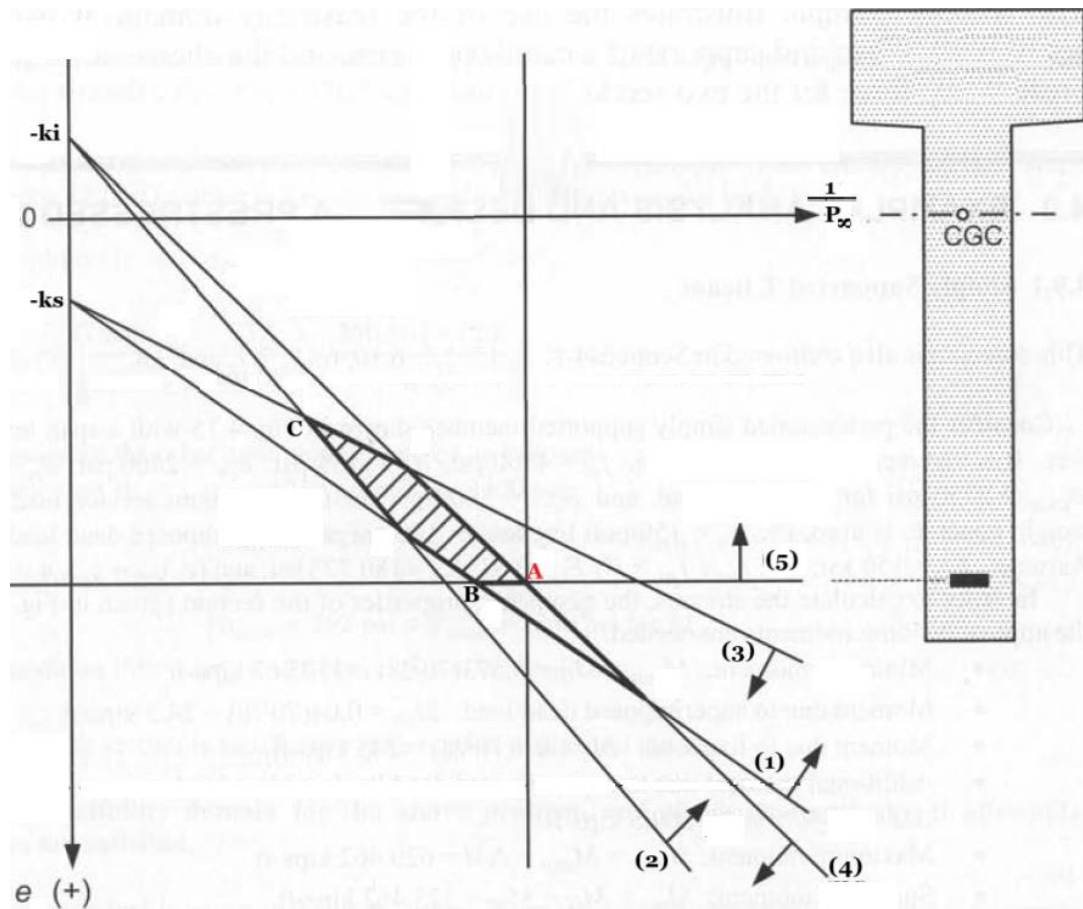


Figura 3.13 - Diagrama de Magnel [3] (adaptado).

No diagrama de Magnel ilustrado na Figura 3.13, o ponto A representa o pré-esforço mínimo à excentricidade máxima, o ponto B o pré-esforço máximo à excentricidade máxima, e o ponto C o pré-esforço máximo admissível, com excentricidade reduzida.

A Tabela 3.8 apresenta o bloco de código utilizado para realizar a combinação de ações e determinar os valores dos momentos fletores nas secções críticas.

Tabela 3.8 - Código para executar a combinação de ações.

```
def combinacao_de_acoes():
    def configurar_modelo():
        ops.wipe()
        ops.model('basic', '-ndm', 2, '-ndf', 3)
        ops.section('Elastic', 1, Ecm, Ac, Iz)
        for i, x in enumerate(nos, start=1):
            ops.node(i, x, 0)
```

```

ops.geomTransf('Linear', 1)
for i in range(1, num_intervalos + 1):
    ops.element('elasticBeamColumn', i, i, i + 1, Ac, Ecm, Iz, 1)
if apoio_1 in apoios:
    ops.fix(nos_id[0] + 1, *apoios[apoio_1])
else:
    print("Atenção, verificar as condições no Apoio da extremidade esquerda.")
    sys.exit()
if apoio_2 in apoios:
    ops.fix(indice_L1 + 1, *apoios[apoio_2])
else:
    print("Atenção, verificar as condições no Apoio interior esquerdo.")
    sys.exit()
if apoio_3 in apoios:
    ops.fix(indice_L2 + 1, *apoios[apoio_3])
else:
    print("Atenção, verificar as condições no Apoio interior direito.")
    sys.exit()
if apoio_4 in apoios:
    ops.fix(nos_id[-1] + 1, *apoios[apoio_4])
else:
    print("Atenção, verificar as condições no Apoio da extremidade direita!")
    sys.exit()
def definir_carregamento(inicio, numero_intervalos, carga):
    ops.timeSeries('Constant', 1)
    ops.pattern('Plain', 1, 1)
    ops.eleLoad('-ele', *range(inicio, numero_intervalos + 1), '-type', '-beamUniform',
carga)
def realizar_analise():
    ops.system('BandSPD')
    ops.numberer('RCM')
    ops.constraints('Plain')
    ops.integrator('LoadControl', 1.0)
    ops.test('NormDispIncr', 1.0e-12, 10, 3)
    ops.algorithm('Newton')
    ops.analysis('Static')
    ops.initialize()
    ok = ops.analyze(1)
def salvar_resultados(displ_file, nmv_file, excel_file):
    ops.printModel('-JSON', '-file', displ_file, '-node')
    ops.printModel('-JSON', '-file', nmv_file, '-ele')

```

```

with open(displ_file, 'r') as file:
    node_disp_data = [line.strip().split() for line in file]
with open(nmv_file, 'r') as file:
    ele_nmv_data = [line.strip().split() for line in file]
node_disp_df = pd.DataFrame(node_disp_data)
ele_nmv_df = pd.DataFrame(ele_nmv_data)
with pd.ExcelWriter(excel_file) as writer:
    node_disp_df.to_excel(writer, sheet_name='Disp', index=False, header=False)
    ele_nmv_df.to_excel(writer, sheet_name='NMV', index=False, header=False)
def plotar_diagrama_momento(titulo, sfacM=5.e-4):
    opsv.section_force_diagram_2d('M', sfacM)
    plt.title(titulo)
    plt.show()
def plotar_modelo_estatico(titulo):
    opsv.plot_model(node_labels=0, element_labels=0)
    plt.title(titulo)
    plt.savefig('modelo_viga_cont3.png')
    plt.show()
def plotar_cargas(titulo):
    opsv.plot_loads_2d(nep=3, sfac=True, fig_wi_he=False, fig_lbrt=False,
fmt_model_loads={'color': 'black', 'linestyle': 'solid', 'linewidth': 1.2, 'marker': '', 'markersize':
1}, node_supports=True, truss_node_offset=0, ax=False)
    plt.title(titulo)
    plt.show()
def extrair_valores_momento(excel_file, sheet_name):
    wb = load_workbook(excel_file, data_only=True)
    CP_NMV = wb[sheet_name]
    valores_coluna_I = []
    valor_inicial = CP_NMV['I7'].value
    if valor_inicial is not None:
        valores_coluna_I.append(valor_inicial)
    indice = 8
    while True:
        celula = f'I{indice}'
        valor = CP_NMV[celula].value
        if valor is None:
            break
        valores_coluna_I.append(valor)
        indice += 8
    return valores_coluna_I
def combinacao_QP_vao():

```

```

configurar_modelo()
definir_carregamento(indice_L1 + 1, indice_L2, psi_2 * q)
realizar_analise()
salvar_resultados('combinacao_qp_vao_Dispatch.txt', 'combinacao_qp_vao_NMV.txt',
'combinacao_qp_vao.xlsx')
#plotar_diagrama_momento('Diagrama de momento fletor (combinação QP - Vão)')
plotar_modelo_estatico('Modelo estático da viga')
#plotar_cargas('Cargas (combinação QP - Vão)')
return extrair_valores_momento('combinacao_qp_vao.xlsx', 'NMV')
def combinacao_QP_apoio():
    configurar_modelo()
    definir_carregamento(1, indice_L2, psi_2 * q)
    realizar_analise()
    salvar_resultados('combinacao_qp_apoio_Dispatch.txt',
'combinacao_qp_apoio_NMV.txt', 'combinacao_qp_apoio.xlsx')
    #plotar_diagrama_momento('Diagrama de momento fletor (combinação QP - Apoio)')
    #plotar_cargas('Cargas (combinação QP - Apoio)')
    return extrair_valores_momento('combinacao_qp_apoio.xlsx', 'NMV')
def combinacao_rara_vao():
    configurar_modelo()
    definir_carregamento(indice_L1 + 1, indice_L2, q)
    realizar_analise()
    salvar_resultados('combinacao_rara_vao_Dispatch.txt', 'combinacao_rara_vao_NMV.txt',
'combinacao_rara_vao.xlsx')
    plotar_diagrama_momento('Diagrama de momento fletor (combinação RARA - Vão)')
    plotar_cargas('Cargas (combinação RARA - Vão)')
    return extrair_valores_momento('combinacao_rara_vao.xlsx', 'NMV')
def combinacao_rara_apoio():
    configurar_modelo()
    definir_carregamento(1, indice_L2, q)
    realizar_analise()
    salvar_resultados('combinacao_rara_apoio_Dispatch.txt',
'combinacao_rara_apoio_NMV.txt', 'combinacao_rara_apoio.xlsx')
    plotar_diagrama_momento('Diagrama de momento fletor (combinação RARA -
Apoio)')
    plotar_cargas('Cargas (combinação RARA - Apoio)')
    return extrair_valores_momento('combinacao_rara_apoio.xlsx', 'NMV')
M_QP_vao_string = combinacao_QP_vao()
M_QP_apoio_string = combinacao_QP_apoio()
M_CR_vao_string = combinacao_rara_vao()
M_CR_apoio_string = combinacao_rara_apoio()

```

```

M_QP_vao = [float(value) for value in M_QP_vao_string]
M_QP_apoio = [float(value) for value in M_QP_apoio_string]
M_CR_vao = [float(value) for value in M_CR_vao_string]
M_CR_apoio = [float(value) for value in M_CR_apoio_string]
return M_QP_vao, M_QP_apoio, M_CR_vao, M_CR_apoio
M_QP_vao, M_QP_apoio, M_CR_vao, M_CR_apoio = combinacao_de_acoes()

```

Com os valores de momentos fletores calculados, procede-se à construção do diagrama de Magnel. Esta versão do TendonTech destina-se a utilizadores com conhecimento prévio em pré-esforço, uma vez que exige a definição prévia do número de cabos. Considera-se o modelo estático da viga analisada, representado na Figura 3.14, sob o carregamento uniformemente distribuído g . Na Tabela 3.9, apresenta-se o código necessário para a importação dos dados utilizados na construção do diagrama de Magnel.

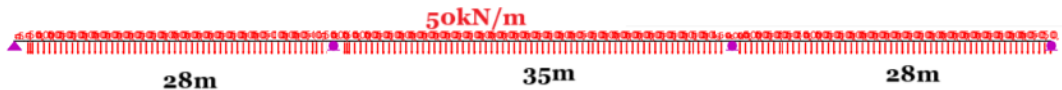


Figura 3.14 - Modelo estático da viga.

Pode estimar-se o valor de momento fletor num dos apoios da seguinte forma:

$$M = \frac{g \cdot L^2}{12} = \frac{50 \cdot 35^2}{12} = 5104 \text{ kNm}$$

$$P_{\infty} \geq \frac{M}{e_{\text{lim}} + k_i} = \frac{5104}{0.889 + 0.249} = 4485 \text{ kN}$$

$$A_p \geq \frac{1,2 \cdot P_{\infty}}{0,8 \cdot f_{pk}} = \frac{1,2 \cdot 4485}{0,8 \cdot 186} = 36 \text{ cm}^2$$

$$n = \frac{A_p}{A_{\text{nominal}}} = \frac{36}{1,4} = 26$$

\therefore assumidos **2 cabos** de 15 cordões

Onde: P_{∞} é o pré-esforço útil; f_{pk} é o valor característico da tensão de rotura do aço de pré-esforço; A_p a área total da secção dos cabos de pré-esforço; A_{nominal} é a área nominal de um cordão; n é o número de cordões.

Tabela 3.9 - Código para importação de dados de entrada do TendonTech: Diagrama de Magnel.

```

from openpyxl import load_workbook

```

```

import matplotlib.pyplot as plt
import numpy as np
from PIL import Image
import sys
wb = load_workbook(r"C:/Users/luton/11. TendonTech 1.0 (BETA)/TT_InputData.xlsx",
data_only=True)
cargas_vigas = wb['cargas_vigas']
n_vaos = int(cargas_vigas['G4'].value) # número de vão pretendidos, [-]
print("\nTendonTech 1.0 2024:\nProjeto de aplicação de pré-esforço\n")
if n_vaos <= 0 or not n_vaos or n_vaos > 4:
    print('\nNa presente versão, apenas são suportadas vigas contínuas com até 4
vãos.\nTendonTech 1.0 2024')
    sys.exit()
cargas_vigas = wb['cargas_vigas']
magnel_positivo = wb['magnel_positivo']
magnel_negativo = wb['magnel_negativo']
perfil_cabo = wb['perfil_cabo']
materiais = wb['materiais']
def dados():
    global fpk, bainha, A_nominal, Ep, n_ETA, \
        vi, vs, zi, zs, ki, ks, e_lim, \
        vi_neg, vs_neg, zi_neg, zs_neg, ki_neg, ks_neg, e_lim_neg, \
        n_cabos, eta
    fck = (materiais['D6'].value) #[MPa]
    fck_ = (materiais['E6'].value) #[MPa]
    fct_o = (materiais['D8'].value) #[MPa]
    fcc_o = (materiais['D9'].value) #[MPa]
    fct_oo = (materiais['D10'].value) #[MPa]
    fcc_oo = (materiais['D11'].value) #[MPa]
    fpk = (materiais['H8'].value) * 10**3 #[kN/m^2]
    bainha = (materiais['H10'].value) #[mm]
    A_nominal = (materiais['H11'].value) * 10**(-4) #[m^2]
    valor_celula = materiais['G14'].value
    # Dividir a string por vírgulas e espaços, e converter cada item para um inteiro
    n_ETA = [int(x.strip()) for x in valor_celula.split(',')] # número de cordões (ETA)
    vi = (magnel_positivo['C12'].value) #[m]
    vs = (magnel_positivo['C13'].value) #[m]
    zi = (magnel_positivo['C14'].value) #[m^3]
    zs = (magnel_positivo['C15'].value) #[m^3]
    ki = (magnel_positivo['C16'].value) #[m]
    ks = (magnel_positivo['C17'].value) #[m]

```

```

e_lim = (magnel_positivo['C18'].value) #[m]
vi_neg = (magnel_negativo['C12'].value) #[m]
vs_neg = (magnel_negativo['C13'].value) #[m]
zi_neg = (magnel_negativo['C14'].value) #[m^3]
zs_neg = (magnel_negativo['C15'].value) #[m^3]
ki_neg = (magnel_negativo['C16'].value) #[m]
ks_neg = (magnel_negativo['C17'].value) #[m]
e_lim_neg = -(magnel_negativo['C18'].value) #[m]
n_cabos = int(perfil_cabo['X7'].value) #[-]
dados()

```

O diagrama de Magnel avalia apenas as condições nas secções críticas. No caso em análise, de uma viga contínua, realizam-se duas avaliações: uma na região de um dos apoios interiores e no meio vão. A Tabela 3.10 apresenta o código para a criação do diagrama de Magnel no meio vão.

Tabela 3.10 - Código para a construção do diagrama de Magnel na secção critica onde o momento fletor é positivo.

```

M_CP_vao = M_fases
M_QP_vao = [a + b for a, b in zip(M_CP_vao, M_QP_vao)]
M_CR_vao = [a + b for a, b in zip(M_CP_vao, M_CR_vao)]
def magnel_vao(ki, ks, eta, fct_o, zs, fcc_o, zi, fcc_oo, fct_oo, e_lim):
    Mmin_vao = max(float(M) for M in M_CP_vao if float(M) > 0) * 10**(-3) # [MNm]
    Mqp_vao = max(float(M) for M in M_QP_vao if float(M) > 0) * 10**(-3) # [MNm]
    Mraro_vao = max(float(M) for M in M_CR_vao if float(M) > 0) * 10**(-3) # [MNm]
    P_oo = sp.symbols('P_oo')
    inequacao_1 = lambda e: e <= -ks + (1/P_oo) * eta * (Mmin_vao - fct_o * zs)
    inequacao_2 = lambda e: e <= -ki + (1/P_oo) * eta * (Mmin_vao + fcc_o * zi)
    inequacao_3 = lambda e: e >= -ks + (1/P_oo) * (Mraro_vao + fcc_oo * zs)
    inequacao_4 = lambda e: e >= -ki + (1/P_oo) * (Mqp_vao - fct_oo * zi)
    inequacao_5 = lambda e: e <= e_lim
    intersec_1_5 = (e_lim + ks) / (eta * (float(Mmin_vao) - fct_o * zs))
    intersec_2_3 = (ki - ks) / (eta * (fcc_o * zi + float(Mmin_vao)) - fcc_oo * zs -
float(Mraro_vao))
    intersec_2_5 = (e_lim + ki) / (eta * (fcc_o * zi + float(Mmin_vao)))
    intersec_3_4 = -(ki - ks) / (fcc_oo * zs + fct_oo * zi - float(Mqp_vao) + float(Mraro_vao))
    intersec_4_5 = (e_lim + ki) / (float(Mqp_vao) - fct_oo * zi) #Ponto A
    if (intersec_4_5) < (intersec_2_5) or (intersec_4_5) < (intersec_1_5):
        print('\nAtenção, aumente o tamanho da secção e/ou a classe de betão.\n')
        sys.exit()

```

```

plt.figure(figsize=(8, 6))
plt.plot([0, intersec_4_5], [0, 0], linestyle='dashed', color='grey', label='eixo neutro')
plt.scatter([0], [-ki], color='yellow', label=f'$k_i = {ki:.3f}m$')
plt.scatter([0], [-ks], color='pink', label=f'$k_s = {ks:.3f}m$')
plt.plot([0, intersec_1_5], [-ks, e_lim], color='blue', label=r'$(1): e \leq -Ks + \frac{1}{P_{00}} \cdot \eta \cdot (M_{\min} - fct_o \cdot Zs)$')
plt.plot([0, intersec_2_5], [-ki, e_lim], color='green', label=r'$(2): e \leq -Ki + \frac{1}{P_{00}} \cdot \eta \cdot (M_{\min} + fcc_o \cdot Zi)$')
plt.plot([0, intersec_3_4], [-ks, 0], color='brown', label=r'$(3): e \geq -Ks + \frac{1}{P_{00}} \cdot (M_{\max\_Raro} + fcc_{00} \cdot Zs)$')
plt.plot([0, intersec_4_5], [-ki, e_lim], color='red', label=r'$(4): e \geq -Ki + \frac{1}{P_{00}} \cdot (M_{\max\_QP} - fct_{00} \cdot Zi)$')
plt.plot([0, intersec_4_5], [e_lim, e_lim], color='grey', label='(5): $e \leq e_{\lim}$')
plt.scatter([intersec_4_5], [e_lim], color='black', label=f'$P_{\infty} = \{1/intersec_4_5:.3f\}MN$')
plt.scatter([intersec_4_5], [e_lim], color='black', label=f'$e_{\lim} = \{e_{\lim:.3f}m$')
plt.title('Diagrama de Magnel - Pré-esforço útil mínimo no vão')
plt.xlabel('$ 1/P_{00}$')
plt.ylabel('e')
plt.gca().invert_yaxis()
plt.grid(True)
plt.legend()
plt.savefig('magnel_vaos.png')
plt.show()
Ap_calc_vao = 1.2 * (intersec_4_5**(-1) * 10**3) / (0.8 * fpk) * 10**4 # [cm^2]
if n_cabos <= 0 or not n_cabos:
    print('Atenção, o número de Cabos não pode ser nulo ou negativo.')
    sys.exit()
else:
    n_vao = round(Ap_calc_vao / (A_nominal * 10**4) / n_cabos)
    if n_vao > n_ETA[-1]:
        print('\nAtenção, aumente número de Cabos e/ou o tamanho da secção.\n')
        sys.exit()
    elif n_vao not in n_ETA:
        n_vao = min([x for x in n_ETA if x > n_vao]) # Ajustamento do número de cordões
por cabo caso não esteja na ETA
Ap_vao = n_vao * A_nominal # [m^2]
Pmax_vao = 0.8 * fpk * Ap_vao # Força de tensionamento por cabo, [kN]
return n_vao, Ap_vao, Pmax_vao

```

O resultado do bloco de código anterior é o gráfico apresentado na Figura 3.15.

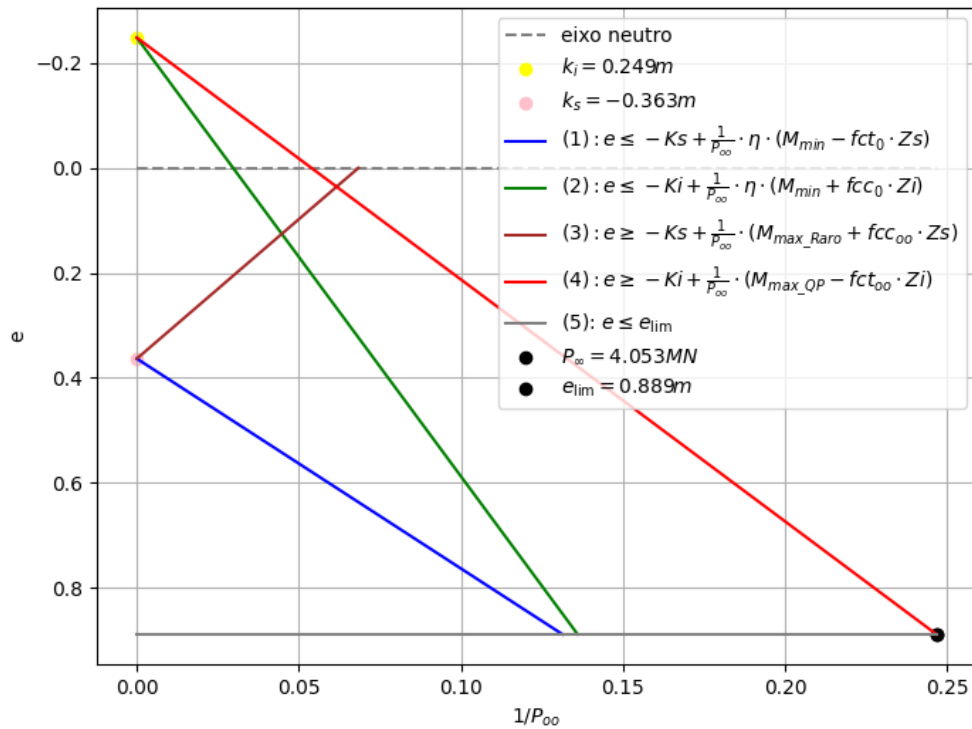


Figura 3.15 - Diagrama de Magnel: pré-esforço útil mínimo nos vãos.

De forma análoga, para a secção crítica sobre os apoios, utiliza-se o mesmo procedimento. A Tabela 3.11 contém o código para a construção do diagrama de Magnel nos apoios, cujo gráfico é apresentado na Figura 3.16.

Tabela 3.11 - Código para a construção do diagrama de Magnel na secção crítica onde o momento fletor é negativo.

```

M_CP_apoio = M_fases
M_QP_apoio = [a + b for a, b in zip(M_CP_apoio, M_QP_apoio)]
M_CR_apoio = [a + b for a, b in zip(M_CP_apoio, M_CR_apoio)]
def magnel_apoio(ki_neg, ks_neg, eta, fct_o, zs_neg, fcc_o, zi_neg, fcc_oo, fct_oo,
e_lim_neg):
    Mmin_apoio = min(float(M) for M in M_CP_apoio if float(M) < 0) * 10**(-3) # [MNm]
    Mqp_apoio = min(float(M) for M in M_QP_apoio if float(M) < 0) * 10**(-3) # [MNm]
    Mraro_apoio = min(float(M) for M in M_CR_apoio if float(M) < 0) * 10**(-3) # [MNm]
    P_oo = sp.symbols('P_oo')
    intersec_1_5 = (e_lim_neg + ks_neg) / (eta * (float(Mmin_apoio) - fct_o * zs_neg))
    intersec_2_3 = (ki_neg - ks_neg) / (eta * (fcc_o * zi_neg + float(Mmin_apoio)) - fcc_oo
* zs_neg - float(Mraro_apoio))
    intersec_2_5 = (e_lim_neg + ki_neg) / (eta * (fcc_o * zi_neg + float(Mmin_apoio)))

```

```

intersec_3_4 = -(ki_neg - ks_neg) / (fcc_oo * zs_neg + fct_oo * zi_neg - float(Mqp_apoio)
+ float(Mraro_apoio))
intersec_4_5 = (e_lim_neg + ki_neg) / (float(Mqp_apoio) - fct_oo * zi_neg) #Ponto A
if (intersec_4_5) < (intersec_2_5) or (intersec_4_5) < (intersec_1_5):
    print('\nAtenção, aumente o tamanho da secção e/ou a classe de betão.\n')
    sys.exit()
plt.figure(figsize=(8, 6))
plt.plot([0, intersec_4_5], [0, 0], linestyle='dashed', color='grey', label='eixo neutro')
plt.scatter([0], [-ki_neg], color='yellow', label=f'$k_i = {ki_neg:.3f}m$')
plt.scatter([0], [-ks_neg], color='pink', label=f'$k_s = {ks_neg:.3f}m$')
plt.plot([0, intersec_1_5], [-ks_neg, e_lim_neg], color='blue', label=r'$(1): e \leq -ks +
\frac{1}{P_{oo}} \cdot \eta \cdot (M_{min} - fct_o \cdot zs)$')
plt.plot([0, intersec_2_5], [-ki_neg, e_lim_neg], color='green', label=r'$(2): e \leq -ki +
\frac{1}{P_{oo}} \cdot \eta \cdot (M_{min} + fcc_o \cdot zi)$')
plt.plot([0, intersec_3_4], [-ks_neg, 0], color='brown', label=r'$(3): e \geq -ks +
\frac{1}{P_{oo}} \cdot (M_{max\_Raro} + fcc_{oo} \cdot zs)$')
plt.plot([0, intersec_4_5], [-ki_neg, e_lim_neg], color='red', label=r'$(4): e \geq -ki +
\frac{1}{P_{oo}} \cdot (M_{max\_QP} - fct_{oo} \cdot zi)$')
plt.plot([0, intersec_4_5], [e_lim_neg, e_lim_neg], color='grey', label='(5): $e \leq
e_{lim}$')
plt.scatter([intersec_4_5], [e_lim_neg], color='black', label=f'$P_{\infty} =
{1/intersec_4_5:.3f}MN$')
plt.scatter([intersec_4_5], [e_lim_neg], color='black', label=f'$e_{lim} =
{e_lim_neg:.3f}m$')
plt.title('Diagrama de Magnel - Pré-esforço útil mínimo no apoio')
plt.xlabel('$ 1/P_{oo}$')
plt.ylabel('e')
plt.gca().invert_yaxis()
plt.grid(True)
plt.legend()
plt.savefig('magnel_apoios.png')
plt.show()
Ap_calc_apoio = 1.2 * (intersec_4_5**(-1) * 10**3) / (0.8 * fpk) * 10**4 # [cm^2]
if n_cabos == 0:
    print('\nAtenção, o número de Cabos não pode ser zero.\n')
    sys.exit()
elif n_cabos < 0:
    print('\nAtenção, o número de Cabos não pode negativo.\n')
    sys.exit()
else:
    n_apoio = round(Ap_calc_apoio / (A_nominal * 10**4) / n_cabos)

```

```

if n_apoio > n_ETA[-1]:
    print('\nAtenção, aumente número de Cabos e/ou o tamanho da secção.\n')
    sys.exit()
elif n_apoio not in n_ETA:
    n_apoio = min([x for x in n_ETA if x > n_apoio])
    Ap_apoio = n_apoio * A_nominal # [m^2]
    Pmax_apoio = 0.8 * fpk * Ap_apoio # [kN]
    return n_apoio, Ap_apoio, Pmax_apoio
n_apoio, Ap_apoio, Pmax_apoio = magnel_apoio(ki_neg, ks_neg, eta, fct_o, zs_neg, fcc_o,
zi_neg, fcc_oo, fct_oo, e_lim_neg)

```

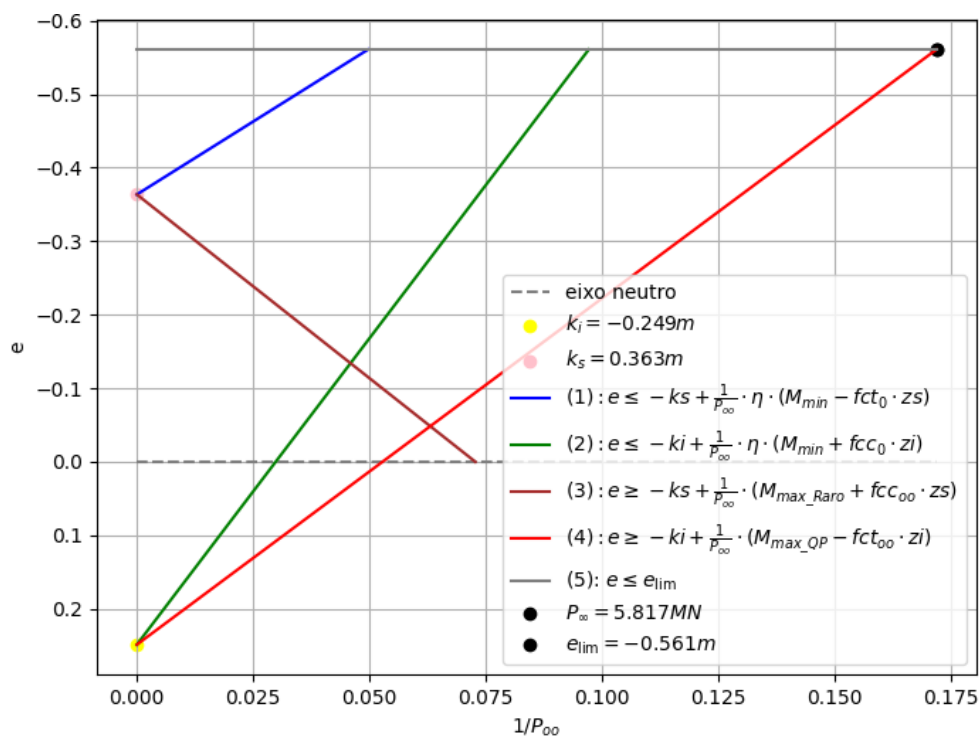


Figura 3.16 - Diagrama de Magnel: pré-esforço útil mínimo nos apoios interiores.

Como se trata de duas situações distintas, a força de tensionamento será definida pelo maior valor entre ambas. A Tabela 3.12 apresenta o bloco de código correspondente para a definição da força de tensionamento. A Figura 3.17 apresenta o resumo dos parâmetros determinados através diagrama de Magnel.

Tabela 3.12 - Código necessário para a definição da força de tensionamento.

```

def forca_tensionamento():
    Pmax = max(Pmax_vao, Pmax_apoio) # [kN]
    n = max(n_vao, n_apoio) # [-]
    Ap = max(Ap_vao, Ap_apoio) # [cm^2]

```

```

if n_cabos >= 1:
    print('Solução (Magnet):\n',
          'Nº Cabos =', n_cabos, 'cabos\n',
          'Nº Cordões por cabo =', n, 'cordões\n',
          'Ap =', round(Ap * 10**4, 3), 'Cm^2\n',
          'Pmax =', round(Pmax, 3), 'kN\n')
    return Pmax, n, Ap
Pmax, n, Ap = forca_tensionamento()

```

```

TendonTech 1.0 2024:
Projeto de aplicação de pré-esforço

Solução (Magnet):
Nº Cabos = 2 cabos
Nº Cordões por cabo = 19 cordões
Ap = 28.5 Cm^2
Pmax = 4240.8 kN

```

Figura 3.17 - Resumo dos parâmetros de pré-esforço determinados pelo diagrama de Magnet.

3.4.3. Modelação geométrica dos cabos

A modelação geométrica dos cabos é realizada de forma simplificada, conforme descrito na Secção 2.3.1 deste documento. Para vigas de um único tramo, utiliza-se a equação (5), enquanto para vigas contínuas aplica-se a equação (6). A Tabela 3.13 apresenta o bloco de código correspondente à definição das equações que descrevem o traçado dos cabos correspondentes ao exemplo em análise.

Tabela 3.13 – Código necessário para a definição das equações que definem o traçado dos cabos.

```

def perfil_cabos(n_cabos):
    if not n_cabos or n_cabos > 10 or n_cabos <= 0:
        print('\nNúmero de cabos não suportado na presente versão!\n(*** TendonTech 1.0
2024 ***)')
        sys.exit()
    else:
        x_c1, x_c2, x_c3, x_c4, x_c5, x_c6, x_c7, x_c8, x_c9, x_c10 = sp.symbols('x_c1:11')
        if n_cabos >= 1:
            global a1_c1, c1_c1, parabola_1_c1, a2_c1, parabola_2_c1, a3_c1, parabola_3_c1,
a4_c1, parabola_4_c1, a5_c1, parabola_5_c1, a6_c1, parabola_6_c1, a7_c1, parabola_7_c1,
a8_c1, parabola_8_c1, a9_c1, parabola_9_c1, a10_c1, c10_c1, parabola_10_c1
            tipo_c1 = [perfil_cabo[f'AD{i}'].value for i in range(14, 24)] # Tipo de traçado
            if tipo_c1[0] == 0 or tipo_c1[0] > 1:

```

```

    print("Troço 1 (cabo 1)\n Traçado não suportado na presente versão!\n(***)
TendonTech 1.0 2024 ***)')
    sys.exit()
elif tipo_c1[0] == 1:
    if perfil_cabo['Y14'].value < -ki or perfil_cabo['Y14'].value > -ks:
        print(f'(Cabo 1)\n Atenção, a excentricidade inicial deve estar dentro dos limites
superior e inferior do núcleo central:\n [ki, ks] = [{-ki:.3f}, {-ks:.3f}])
        sys.exit()
    elif perfil_cabo['Y14'].value >= 0:
        a1_c1 = - (abs(perfil_cabo['Y14'].value) + e_lim) / H1**2
    elif perfil_cabo['Y14'].value < 0:
        a1_c1 = - ( - perfil_cabo['Y14'].value + e_lim) / H1**2
    c1_c1 = perfil_cabo['Y14'].value
    parabola_1_c1 = a1_c1 * x_c1**2 + c1_c1
if tipo_c1[1] == 0 or tipo_c1[1] > 1:
    print("Troço 2 (cabo 1)\n Traçado não suportado na presente versão!\n(***)
TendonTech 1.0 2024 ***)')
    sys.exit()
elif tipo_c1[1] == 1:
    K2_c1 = (0.1) / (0.5 + 0.1) * (e_lim + abs(e_lim_neg)) + e_lim_neg #m
    a2_c1 = - (abs(K2_c1) + e_lim) / H2**2
    parabola_2_c1 = a2_c1 * x_c1**2
if tipo_c1[2] == 0 or tipo_c1[2] > 1:
    print("Troço 3 (cabo 1)\n Traçado não suportado na presente versão!\n(***)
TendonTech 1.0 2024 ***)')
    sys.exit()
elif tipo_c1[2] == 1:
    a3_c1 = ((0.1) / (0.5 + 0.1) * (e_lim + abs(e_lim_neg))) / H3**2
    parabola_3_c1 = a3_c1 * x_c1**2
if tipo_c1[3] == 0 or tipo_c1[3] > 1:
    print("Troço 4 (cabo 1)\n Traçado não suportado na presente versão!\n(***)
TendonTech 1.0 2024 ***)')
    sys.exit()
elif tipo_c1[3] == 1:
    K4_c1 = (0.2) / (0.3 + 0.2) * (e_lim + abs(e_lim_neg)) #m
    a4_c1 = K4_c1 / H4**2
    parabola_4_c1 = a4_c1 * x_c1**2
if tipo_c1[4] == 0 or tipo_c1[4] > 1:
    print("Troço 5 (cabo 1)\n Traçado não suportado na presente versão!\n(***)
TendonTech 1.0 2024 ***)')
    sys.exit()

```

```

elif tipo_c1[4] == 1:
    K5_c1 = e_lim + abs(e_lim_neg) - K4_c1 #m
    a5_c1 = - K5_c1 / H5**2
    parabola_5_c1 = a5_c1 * x_c1**2
if tipo_c1[5] == 0 or tipo_c1[5] > 1:
    print("Troço 6 (cabo 1)\n Traçado não suportado na presente versão!\n(***)
TendonTech 1.0 2024 ***)')
    sys.exit()
elif tipo_c1[5] == 1:
    K6_c1 = (0.1) / (0.4 + 0.1) * (e_lim + abs(e_lim_neg)) + e_lim_neg #m
    a6_c1 = - (abs(K6_c1) + e_lim) / H6**2
    parabola_6_c1 = a6_c1 * x_c1**2
if tipo_c1[6] == 0 or tipo_c1[6] > 1:
    print("Troço 7 (cabo 1)\n Traçado não suportado na presente versão!\n(***)
TendonTech 1.0 2024 ***)')
    sys.exit()
elif tipo_c1[6] == 1:
    a7_c1 = ((0.1) / (0.4 + 0.1) * (e_lim + abs(e_lim_neg))) / H7**2
    parabola_7_c1 = a7_c1 * x_c1**2
if tipo_c1[7] == 0 or tipo_c1[7] > 1:
    print("Troço 8 (cabo 1)\n Traçado não suportado na presente versão!\n(***)
TendonTech 1.0 2024 ***)')
    sys.exit()
elif tipo_c1[7] == 1:
    K8_c1 = (0.2) / (0.3 + 0.2) * (e_lim + abs(e_lim_neg)) #m
    a8_c1 = K8_c1 / H8**2
    parabola_8_c1 = a8_c1 * x_c1**2
if tipo_c1[8] == 0 or tipo_c1[8] > 1:
    print("Troço 9 (cabo 1)\n Traçado não suportado na presente versão!\n(***)
TendonTech 1.0 2024 ***)')
    sys.exit()
elif tipo_c1[8] == 1:
    K9_c1 = e_lim + abs(e_lim_neg) - K8_c1 #m
    a9_c1 = - K9_c1 / H9**2
    parabola_9_c1 = a9_c1 * x_c1**2
if tipo_c1[9] == 0 or tipo_c1[9] > 1:
    print("Troço 10 (cabo 1)\n Traçado não suportado na presente versão!\n(***)
TendonTech 1.0 2024 ***)')
    sys.exit()
elif tipo_c1[9] == 1:
    if perfil_cabo['AB23'].value < - ki or perfil_cabo['AB23'].value > -ks:

```

```

        print(f(cabo 1)\n Atenção, a excentricidade final deve estar dentro dos limites
superior e inferior do núcleo central:\n [ki, ks] = [{-ki:.3f}, {-ks:.3f}])
        sys.exit()
    elif perfil_cabo['AB23'].value >= 0:
        a10_c1 = - (abs(perfil_cabo['AB23'].value) + e_lim) / H10**2
    elif perfil_cabo['AB23'].value < 0:
        a10_c1 = - ( - perfil_cabo['AB23'].value + e_lim) / H10**2
    c10_c1 = perfil_cabo['AB23'].value
    parabola_10_c1 = a10_c1 * x_c1**2 + c10_c1
    perfil_cabos(n_cabos)

```

No caso de uma viga simples, executa-se o bloco de código apresentado na Tabela 3.14. Para uma viga contínua com dois vãos, é executado o bloco de código descrito na Tabela 3.15, com a visualização em CAD ilustrada na Figura 3.18.

Tabela 3.14 – Código necessário para a visualização da geometria dos cabos no caso de uma viga simples.

```

def coordenadas_cabos(nos):
    y_c1 = [a_c1 * i**2 + b_c1 * i + c_c1 for i in nos]
    y_c2 = [a_c2 * i**2 + b_c2 * i + c_c2 for i in nos]
    coord = pd.DataFrame({
        'x [m]': nos,
        'y (cabo 1) [m]': y_c1,
        'y (cabo 2) [m]': y_c2
    })
    coord.to_excel("TT_Coordenadas.xlsx", index=False)
coord = coordenadas_cabos(nos)
def desenhar_cabo(msp, L, vi, vs, offset_x, x_valores, y_valores):
    canto_esq = (offset_x, -vi)
    canto_dir = (offset_x + L, -vi)
    topo_dir = (offset_x + L, -vs)
    topo_esq = (offset_x, -vs)
    msp.add_line(canto_esq, canto_dir)
    msp.add_line(canto_dir, topo_dir)
    msp.add_line(topo_dir, topo_esq)
    msp.add_line(topo_esq, canto_esq)
    for x, y in zip(x_valores, y_valores):
        msp.add_point((x + offset_x, -y))
    for i in range(len(x_valores) - 1):
        msp.add_line(
            (x_valores[i] + offset_x, -y_valores[i]),

```

```

        (x_valores[i + 1] + offset_x, -y_valores[i + 1])
    )
    doc = ezdxf.new()
    msp = doc.modelspace()
    distancia_entre_cabos = 2
    desenhar_cabo(msp, L, vi, vs, 0, x_valores_c1, y_valores_c1)
    desenhar_cabo(msp, L, vi, vs, L + distancia_entre_cabos, x_valores_c2,
y_valores_c2)
    doc.saveas("TT_Cabos.dxf")

```

Tabela 3.15 – Código necessário para a visualização do traçado dos cabos no caso de uma viga contínua com dois vãos.

```

def coordenadas_cabos(nos):
    y1_c1 = [parabola_1_c1(i) for i in range(len(nos[:indice_H1 + 1]))]
    y2_c1 = [parabola_2_c1(i) for i in range(len(nos[indice_H1:indice_H2 + 1]))]
    y3_c1 = [parabola_3_c1(i) for i in range(len(nos[indice_H2:indice_L1 + 1]))]
    y4_c1 = [parabola_4_c1(i) for i in range(len(nos[indice_L1:indice_H4 + 1]))]
    y5_c1 = [parabola_5_c1(i) for i in range(len(nos[indice_H4:indice_H5 + 1]))]
    y6_c1 = [parabola_6_c1(i) for i in range(len(nos[indice_H5:indice_L2 + 1]))][::-1]
    excentricidades_c1 = y1_c1[:-1] + y2_c1[:-1] + y3_c1[:-1] + y4_c1[:-1] + y5_c1[:-1] +
y6_c1
    y1_c2 = [parabola_1_c2(i) for i in range(len(nos[:indice_H1 + 1]))]
    y2_c2 = [parabola_2_c2(i) for i in range(len(nos[indice_H1:indice_H2 + 1]))]
    y3_c2 = [parabola_3_c2(i) for i in range(len(nos[indice_H2:indice_L1 + 1]))]
    y4_c2 = [parabola_4_c2(i) for i in range(len(nos[indice_L1:indice_H4 + 1]))]
    y5_c2 = [parabola_5_c2(i) for i in range(len(nos[indice_H4:indice_H5 + 1]))]
    y6_c2 = [parabola_6_c2(i) for i in range(len(nos[indice_H5:indice_L2 + 1]))][::-1]
    excentricidades_c2 = y1_c2[:-1] + y2_c2[:-1] + y3_c2[:-1] + y4_c2[:-1] + y5_c2[:-1] +
y6_c2
    coord = pd.DataFrame({
        'x [m]': nos,
        'y (cabo 1) [m]': excentricidades_c1,
        'y (cabo 2) [m]': excentricidades_c2
    })
    coord.to_excel("TT_Coordenadas.xlsx", index=False)
    return coord, excentricidades_c1, excentricidades_c2
def desenhar_segmentos(msp, pontos, closed=False):
    msp.add_lwpolyline(pontos, close=closed)
def desenhar_cabo(msp, L, vi, vs, offset_x, nos, excentricidades, c):

```

```

viga = [(offset_x, -vi), (offset_x + L, -vi), (offset_x + L, -vs), (offset_x, -vs)]
msp.add_lwpolyline(viga, close=True)
segmentos = [
    (slice(None, indice_H1 + 1), lambda x, y: (x + offset_x, y - e_lim - eval(f'c1_{c}')),
True),
    (slice(indice_H1, indice_H2 + 1), lambda x, y: (x + offset_x, -y), False),
    (slice(indice_H2, indice_L1 + 1), lambda x, y: (x + offset_x, y - e_lim_neg - K2),
True),
    (slice(indice_L1, indice_H4 + 1), lambda x, y: (x + offset_x, -y), False),
    (slice(indice_H4, indice_H5 + 1), lambda x, y: (x + offset_x, y - K5 - e_lim), True),
    (slice(indice_H5, None), lambda x, y: (x + offset_x, y - eval(f'c6_{c}') - e_lim), True)
]
for segmento, transform, reverse in segmentos:
    pontos = [transform(x, y) for x, y in zip(nos[segmento][::-1 if reverse else 1]),
excentricidades[segmento]]
    desenhar_segmentos(msp, pontos)
def main():
    try:
        doc = ezdxf.new('R2010')
        msp = doc.modelspace()
        desenhar_cabo(msp, L, vi, vs, 0, nos, excentricidades_c1, 'c1')
        desenhar_cabo(msp, L, vi, vs, L + 2, nos, excentricidades_c2, 'c2')
        filename = 'TT_Cabos.dxf'
        doc.saveas(filename)
        print(f"\nFicheiro para visualização do traçado dos cabos salvo como:
{filename}\n")
    except Exception as e:
        print(f'Ocorreu um erro: {e}')
if __name__ == '__main__':
    coord, excentricidades_c1, excentricidades_c2 = coordenadas_cabos(nos)
    main()

```

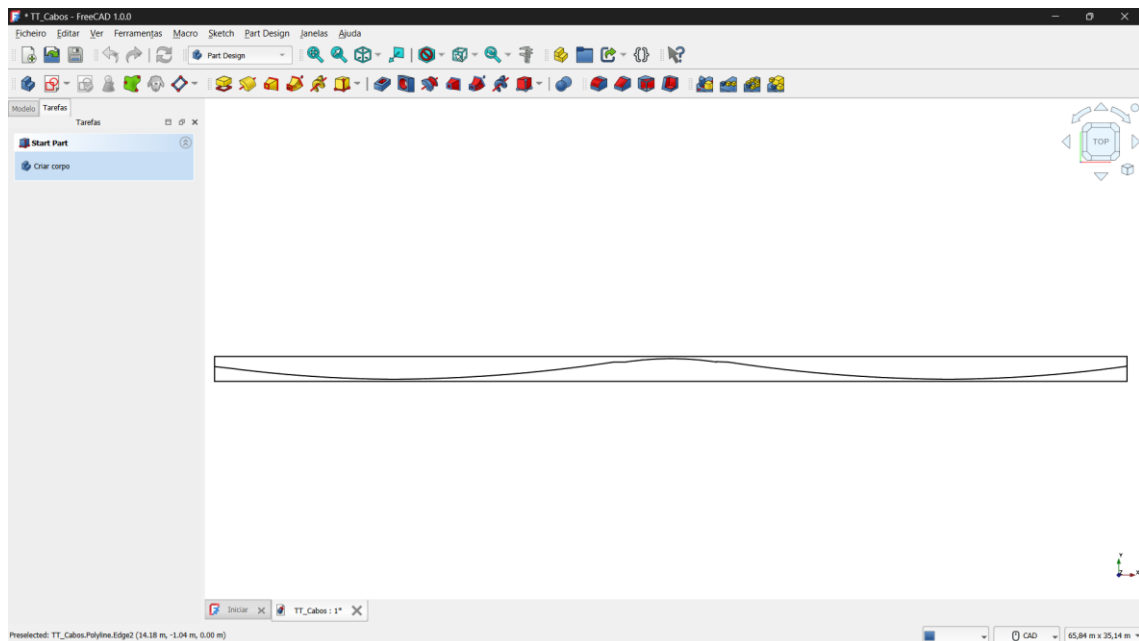


Figura 3.18 – Visualização do traçado dos cabos em CAD, para uma viga contínua com dois vãos.

3.4.4. Quantificação das perdas imediatas

A quantificação das perdas segue a norma NP EN 1992-1-1 [5], conforme detalhado na Secção 2.5 deste documento. Os blocos de código correspondentes estão alinhados com o exemplo ilustrado na Figura 3.5. A Figura 2.13 apresenta um resumo dos tipos de perdas consideradas, sendo as perdas por atrito calculadas com base na Expressão (9) e apresenta-se na Tabela 3.16 o bloco de código necessário para calcular a soma dos desvios angulares ao longo da viga.

Tabela 3.16 - Código necessário para calcular a soma dos desvios angulares ao longo do comprimento da viga.

```
def perdas_imediatas(n_estic, n_cabos):
    if not n_estic or n_estic > 2 or n_estic <= 1:
        print('Atenção, são requeridos pelo menos 2 esticamentos.')
        sys.exit()
    else:
        if not n_cabos or n_cabos > 10 or n_cabos <= 0:
            print('\nNúmero de cabos não suportado na presente versão!\n(** TendonTech 1.0
2024 **)')
            sys.exit()
        if n_cabos >= 1:
            global perdas_mu_c1, along_c1, perdas_sl_c1, along_2_c1, Pmo_C1
            def calcular_desvios_angulares_c1(a, H):
                x_values = np.linspace(0, H, num=int(H)+1)
                return [round(math.atan(2 * abs(a) * x), 4) for x in x_values]
```

```

def soma_cumulativa_c1(angulo_anterior, novos_angulos):
    return [round(angulo_anterior + angulo, 4) for angulo in novos_angulos]
angulos1_c1 = calcular_desvios_angulares_c1(a1_c1, H1)
angulos2_c1 = soma_cumulativa_c1(angulos1_c1[-1],
calcular_desvios_angulares_c1(a2_c1, H2))
angulos3_c1 = soma_cumulativa_c1(angulos2_c1[-1],
calcular_desvios_angulares_c1(a3_c1, H3))
angulos4_c1 = soma_cumulativa_c1(angulos3_c1[-1],
calcular_desvios_angulares_c1(a4_c1, H4))
angulos5_c1 = soma_cumulativa_c1(angulos4_c1[-1],
calcular_desvios_angulares_c1(a5_c1, H5))
angulos6_c1 = soma_cumulativa_c1(angulos5_c1[-1],
calcular_desvios_angulares_c1(a6_c1, H6))
angulos7_c1 = soma_cumulativa_c1(angulos6_c1[-1],
calcular_desvios_angulares_c1(a7_c1, H7))
angulos8_c1 = soma_cumulativa_c1(angulos7_c1[-1],
calcular_desvios_angulares_c1(a8_c1, H8))
angulos9_c1 = soma_cumulativa_c1(angulos8_c1[-1],
calcular_desvios_angulares_c1(a9_c1, H9))
angulos10_c1 = soma_cumulativa_c1(angulos9_c1[-1],
calcular_desvios_angulares_c1(a10_c1, H10))
desv_angulares_c1 = angulos1_c1 + angulos2_c1 + angulos3_c1 + angulos4_c1 +
angulos5_c1 + angulos6_c1 + angulos7_c1 + angulos8_c1 + angulos9_c1 + angulos10_c1
if len(desv_angulares_c1) > len(nos):
    desv_angulares_c1 = desv_angulares_c1[:len(nos)]
elif len(desv_angulares_c1) < len(nos):
    desv_angulares_c1.extend([(desv_angulares_c1[-1]] * (len(nos) -
len(desv_angulares_c1))))
angulos_c1 = [desv_angulares_c1[i] - desv_angulares_c1[i + 1] for i in range(1,
len(desv_angulares_c1) - 1)]

```

Após calcular a soma dos desvios angulares ao longo do comprimento da viga (θ) e importar os valores do coeficiente de atrito (μ) e do desvio angular parasita (k) da base de dados, é possível estimar a perda de pré-esforço por atrito ao longo da viga. A Tabela 3.17 apresenta o bloco de código correspondente a este cálculo.

Tabela 3.17 - Código para quantificar as perdas de pré-esforço por atrito.

```

data_mu_c2 = {
    'Secção': seccoes,
    'x [m]': nos,

```

```

        'Alfa [rad]': desv_angulares_c2
    }
    data_mu_c2['dPmu [kN]'] = [round(Pmax * (1 - math.exp(-mu * (alfa + k * x)))) for x, alfa in zip(data_mu_c2['x [m]'], data_mu_c2['Alfa [rad]'])]
    data_mu_c2['Pmu [kN]'] = [round(Pmax * math.exp(-mu * (alfa + k * x))) for x, alfa in zip(data_mu_c2['x [m]'], data_mu_c2['Alfa [rad]'])]
    data_mu_c2['d_mu [%]'] = [round((dPmu / Pmax) * 100, 2) for dPmu in data_mu_c2['dPmu [kN]']]
    perdas_atrito_c2 = pd.DataFrame(data_mu_c2)
    data_mu2_c2 = {
        'Secção': data_mu_c2['Secção'],
        'x [m]': data_mu_c2['x [m]'],
        'Alfa [rad]': data_mu_c2['Alfa [rad]']
    }
    data_mu2_c2['dPmu [kN]'] = [round(Pmax * (1 - math.exp(-mu * (alfa + k * x)))) for x, alfa in zip(data_mu2_c2['x [m]'][:-1], data_mu2_c2['Alfa [rad]'][:-1])]
    data_mu2_c2['Pmu [kN]'] = [round(Pmax * math.exp(-mu * (alfa + k * x))) for x, alfa in zip(data_mu2_c2['x [m]'][:-1], data_mu2_c2['Alfa [rad]'][:-1])]
    data_mu2_c2['d_mu [%]'] = [round((dPmu / Pmax) * 100, 2) for dPmu in data_mu2_c2['dPmu [kN]']]
    perdas_atrito2_c2 = pd.DataFrame(data_mu2_c2)
    print('\nCABO II: Perdas por Atrito')
    data_mu_mu2_c2 = {
        'Secção': seccoes,
        'x [m]': nos,
        'Alfa [rad]': desv_angulares_c2,
        'dPmu [kN]': data_mu_c2['dPmu [kN]'][:len(perdas_atrito_c2['x [m]']) // 2] + data_mu2_c2['dPmu [kN]'][len(perdas_atrito2_c2['x [m]']) // 2:],
        'Pmu [kN]': data_mu_c2['Pmu [kN]'][:len(perdas_atrito_c2['x [m]']) // 2] + data_mu2_c2['Pmu [kN]'][len(perdas_atrito2_c2['x [m]']) // 2:],
        'd_mu [%]': data_mu_c2['d_mu [%]'][:len(perdas_atrito_c2['x [m]']) // 2] + data_mu2_c2['d_mu [%]'][len(perdas_atrito2_c2['x [m]']) // 2:],
    }
    perdas_mu_c2 = pd.DataFrame(data_mu_mu2_c2)
    print(perdas_mu_c2)
    plt.figure(figsize=(20, 5))
    plt.plot(perdas_atrito_c2['x [m]'], perdas_atrito_c2['Pmu [kN]'], marker='o', label='1º Esticamento')
    plt.plot(perdas_atrito2_c2['x [m]'][len(perdas_atrito2_c2['x [m]']) // 2:], perdas_atrito2_c2['Pmu [kN]'][len(perdas_atrito2_c2['x [m]']) // 2:], marker='o', linestyle='-', label='2º Esticamento')

```

```

plt.title('CABO II: Pré-esforço após perdas por atrito,  $\Delta P_{\mu}$ ')
plt.xlabel('Comprimento da viga, L [m]')
plt.ylabel('$P_{\mu}$ [kN]')
plt.grid(True)
plt.legend()
plt.show()

```

O bloco de código anterior gera como resultado uma tabela exibida diretamente na consola, conforme indicado na Figura 3.19, e um gráfico representado na Figura 3.20.

Secção	x [m]	Alfa [rad]	dPmu [kN]	Pmu [kN]	d_mu [%]	
0	S1	0.0	0.0000	0	4241	0.00
1	S2	1.0	0.0144	21	4220	0.50
2	S3	2.0	0.0289	41	4200	0.97
3	S4	3.0	0.0433	62	4179	1.46
4	S5	4.0	0.0577	82	4159	1.93
...
92	S93	87.0	1.6462	82	4159	1.93
93	S94	88.0	1.6552	62	4179	1.46
94	S95	89.0	1.6642	41	4200	0.97
95	S96	90.0	1.6732	21	4220	0.50
96	S97	91.0	1.6821	0	4241	0.00

Figura 3.19 - Perdas de pré-esforço por atrito.

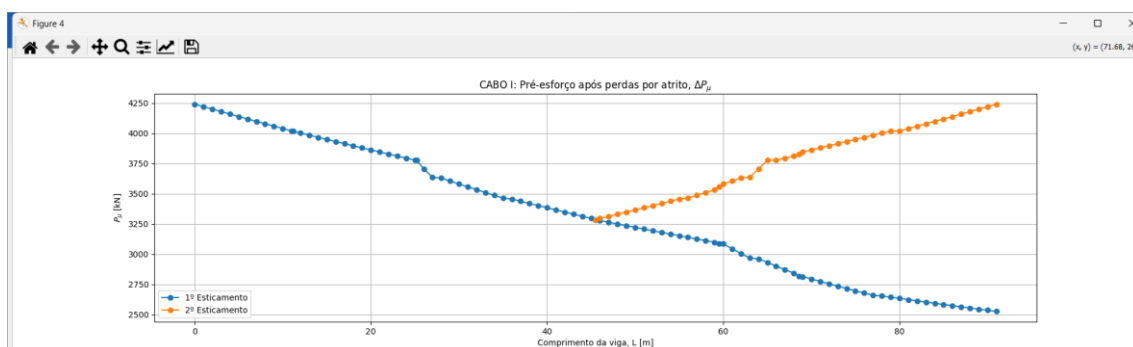


Figura 3.20 - Variação das perdas de pré-esforço por ao atrito.

Para quantificar as perdas causadas pela reentrada das cunhas nas ancoragens, utiliza-se as equações (12) e (13). Inicialmente, estima-se o comprimento afetado por essa reentrada, considerando o deslocamento do sistema especificado na ETA [13]. O cálculo envolve a área total de aço de pré-esforço previamente determinada, o módulo de elasticidade do aço e a perda por

metro linear resultante do atrito. A variação das perdas devido à reentrada das cunhas deve apresentar uma configuração semelhante à ilustrada na Figura 2.15.

Apresenta-se na Tabela 3.18 o excerto de código necessário para estimar as perdas devidas à reentrada das cunhas.

Tabela 3.18 - Código necessário para estimar as perdas devidas a reentrada das cunhas nas ancoragens.

```

coeficientes_c2 = np.polyfit(data_mu_c2['x [m]'], data_mu_c2['Pmu [kN]'], 1)
i_c2 = abs(coeficientes_c2[0]) # [kN/m]
Lambda_c2 = np.sqrt((Ep * Ap * dsl / i_c2)) # [m]
data_sl_c2 = {
    'Secção': data_mu_c2['Secção'],
    'x [m]': data_mu_c2['x [m]'],
    'Pmu [kN]': data_mu_c2['Pmu [kN]'],
}
data_sl_c2['dPsl [kN]'] = [round(2 * (Lambda_c2 - x) * i_c2) if (Lambda_c2 - x) > 0
else 0 for x in data_sl_c2['x [m]']]
data_sl_c2['Psl [kN]'] = [round(P - 2 * (Lambda_c2 - x) * i_c2) if (Lambda_c2 - x) >
0 else P for x, P in zip(data_sl_c2['x [m]'], data_sl_c2['Pmu [kN]'])]
data_sl_c2['d_sl [%]'] = [round(d_mu + (dPsl / Pmax) * 100, 2) for d_mu, dPsl in
zip(data_mu_c2['d_mu [%]'], data_sl_c2['dPsl [kN]'])]
reentrada_cunhas_c2 = pd.DataFrame(data_sl_c2)
omega_c2 = simpson(reentrada_cunhas_c2['Psl [kN]'], reentrada_cunhas_c2['x
[m]'])
along_c2 = np.sqrt(abs(omega_c2) / (Ep * Ap)) # alongamento teórico [m]
print("\nCABO II: Alongamento (1º esticamento) =", round(along_c2 * 10**3, 3),
'mm')
data_sl2_c2 = {
    'Secção': data_mu2_c2['Secção'],
    'x [m]': data_mu2_c2['x [m]'],
    'Pmu [kN]': data_mu2_c2['Pmu [kN]'],
}
data_sl2_c2['dPsl [kN]'] = [round(2 * (Lambda_c2 - x) * i_c2) if (Lambda_c2 - x) >
0 else 0 for x in data_sl2_c2['x [m]'][::-1]]
data_sl2_c2['Psl [kN]'] = [round(P - 2 * (Lambda_c2 - x) * i_c2) if (Lambda_c2 - x)
> 0 else P for x, P in zip(data_sl2_c2['x [m]'][::-1], data_sl2_c2['Pmu [kN]'])]
data_sl2_c2['d_sl [%]'] = [round(d_mu + (dPsl / Pmax) * 100, 2) for d_mu, dPsl in
zip(data_mu2_c2['d_mu [%]'], data_sl2_c2['dPsl [kN]'])]
reentrada_cunhas2_c2 = pd.DataFrame(data_sl2_c2)
print("\nCABO II: Perdas por reentrada das cunhas")
data_sl_sl2_c2 = {

```

```

'Secção': seccoes,
'x [m]': nos,
'dPsl [kN]': data_sl_c2['dPsl [kN]'][:len(reentrada_cunhas_c2['x [m]']) // 2] +
data_sl2_c2['dPsl [kN]'][:len(reentrada_cunhas_c2['x [m]']) // 2:],
'Psl [kN]': data_sl_c2['Psl [kN]'][:len(reentrada_cunhas_c2['x [m]']) // 2] +
data_sl2_c2['Psl [kN]'][:len(reentrada_cunhas_c2['x [m]']) // 2:],
'd_sl [%]': data_sl_c2['d_sl [%]'][:len(reentrada_cunhas_c2['x [m]']) // 2] +
data_sl2_c2['d_sl [%]'][:len(reentrada_cunhas_c2['x [m]']) // 2:],
}
perdas_sl_c2 = pd.DataFrame(data_sl_sl2_c2)
print(perdas_sl_c2)
x2 = reentrada_cunhas_c2['x [m]'][:len(reentrada_cunhas_c2['x [m]']) // 2:]
y2_1 = reentrada_cunhas_c2['Psl [kN]'][:len(reentrada_cunhas_c2['x [m]']) // 2:]
y2_2 = reentrada_cunhas2_c2['Psl [kN]'][:len(reentrada_cunhas2_c2['x [m]']) // 2:]
integral1_c2 = simpson(y2_1, x2)
integral2_c2 = simpson(y2_2, x2)
omega_2_c2 = integral1_c2 + integral2_c2
#omega_2_c2 = simpson(reentrada_cunhas_c2['x
[m]'][:len(reentrada_cunhas_c2['x [m]']) // 2:], reentrada_cunhas_c2['Psl
[kN]'][:len(reentrada_cunhas_c2['x [m]']) // 2:], reentrada_cunhas2_c2['Psl
[kN]'][:len(reentrada_cunhas2_c2['x [m]']) // 2:])
along_2_c2 = np.sqrt(abs(omega_2_c2) / (Ep * Ap))
print('\nCABO II: Alongamento (2º esticamento) =', round(along_2_c2 * 10**3, 3),
'mm')
plt.figure(figsize=(20, 5))
plt.plot(reentrada_cunhas_c2['x [m]'], reentrada_cunhas_c2['Psl [kN]'], marker='o',
linestyle='-', label='1º Esticamento')
plt.plot(reentrada_cunhas2_c2['x [m]'][:len(reentrada_cunhas2_c2['x [m]']) // 2:],
reentrada_cunhas2_c2['Psl [kN]'][:len(reentrada_cunhas2_c2['x [m]']) // 2:], marker='o',
linestyle='-', label='2º Esticamento')
plt.fill_between(reentrada_cunhas_c2['x [m]'][:len(reentrada_cunhas_c2['x [m]']) //
2:], reentrada_cunhas_c2['Psl [kN]'][:len(reentrada_cunhas_c2['x [m]']) // 2:],
reentrada_cunhas2_c2['Psl [kN]'][:len(reentrada_cunhas2_c2['x [m]']) // 2:], color='gray',
alpha=0.1)
plt.title('CABO II: Pré-esforço após perdas por reentrada das cunhas,  $\Delta P_{sl}$ ')
plt.xlabel('Comprimento da viga, L [m]')
plt.ylabel('$P_{sl}$ [kN]')
plt.legend()
plt.grid(True)
plt.show()

```

O código na Tabela 3.18 também gera as estimativas de dois alongamentos teóricos correspondentes aos dois tensionamentos analisados, além de uma tabela com os valores das perdas em cada secção, conforme apresentado na Figura 3.22. O gráfico correspondente é ilustrado na Figura 3.21.

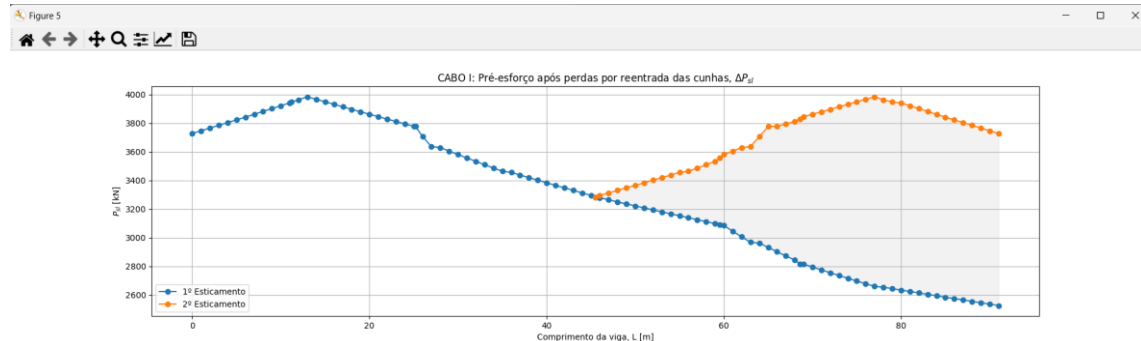


Figura 3.21 – Variação das perdas devidas a reentrada das cunhas nas ancoragens.

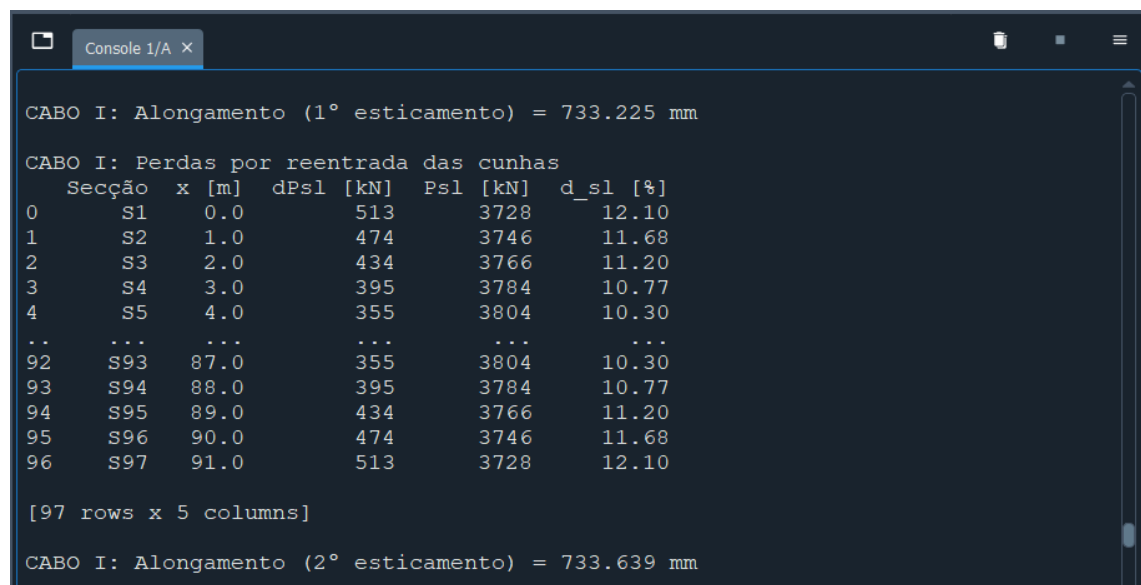


Figura 3.22 - Perdas devidas à reentrada das cunhas nas ancoragens e alongamentos teóricos.

Com as perdas devidas ao atrito e à reentrada das cunhas já determinadas, procede-se à estimativa das cargas equivalentes ao pré-esforço. Esses valores são necessários para calcular as perdas decorrentes da deformação elástica do betão no cabo 1, causada pelo tensionamento do cabo 2. O código correspondente é apresentado na Tabela 3.19.

Tabela 3.19 - Código necessário para estimar os esforços equivalentes ao pré-esforço na viga.

```

ops.wipe()
ops.model('basic', '-ndm', 2, '-ndf', 3)
ops.section('Elastic', 1, Ecm, Ac, Iz)
  
```

```

for i, x in enumerate(nos, start=1):
    ops.node(i, x, 0)
ops.geomTransf('Linear', 1)
for i in range(1, num_intervalos + 1):
    ops.element('elasticBeamColumn', i, i + 1, Ac, Ecm, Iz, 1)
if apoio_1 in apoios:
    ops.fix(nos_id[0] + 1, *apoios[apoio_1])
else:
    print("Atenção, verificar as condições no Apoio da extremidade esquerda.")
    sys.exit()
if apoio_2 in apoios:
    ops.fix(indice_L1 + 1, *apoios[apoio_2])
else:
    print("Atenção, verificar as condições no Apoio interior esquerdo.")
    sys.exit()
if apoio_3 in apoios:
    ops.fix(indice_L2 + 1, *apoios[apoio_3])
else:
    print("Atenção, verificar as condições no Apoio interior direito.")
    sys.exit()
if apoio_4 in apoios:
    ops.fix(nos_id[-1] + 1, *apoios[apoio_4])
else:
    print("Atenção, verificar as condições no Apoio da extremidade direita!")
    sys.exit()
ops.timeSeries('Constant', 1)
ops.pattern('Plain', 1, 1)
ops.system('BandSPD')
ops.numberer('RCM')
ops.constraints('Plain')
ops.integrator('LoadControl', .1)
ops.test('NormDispIncr', 1.0e-12, 10, 3)
ops.algorithm('Newton')
ops.analysis('Static')
for i, (P, alfa) in enumerate(zip(perdas_sl_c2['Psl [kN]'], angulos_c2), start=2):
    if i <= indice_H2:
        ops.load(i, 0., (float(P) * float(abs(np.sin(alfa))))), 0.)
    elif i >= indice_H4 and i <= indice_H6:
        ops.load(i, 0., (float(P) * float(abs(np.sin(alfa))))), 0.)
    elif i >= indice_H8:
        ops.load(i, 0., (float(P) * float(abs(np.sin(alfa))))), 0.)

```

```

else:
    ops.load(i, o., (-float(P) * float(abs(np.sin(alfa))))), o.)
ops.initialize()
ok = ops.analyze(1)
#opsv.plot_loads_2d(nep=3, sfac=True, fig_wi_he=False, fig_lbrt=False,
fmt_model_loads={'color': 'black', 'linestyle': 'solid', 'linewidth': 1.2, 'marker': '', 'markersize':
1}, node_supports=True, truss_node_offset=0, ax=False)
#plt.title('CABO II - Cargas equivalentes de pré-esforço')
#opsv.section_force_diagram_2d('M', 5.e-4)
#plt.title('CABO II - Momentos devidos ao pré-esforço')
ops.initialize()
ok = ops.analyze(1)
ops.printModel('-JSON', '-file', 'Po_Displacement.txt', '-node')
ops.printModel('-JSON', '-file', 'Po_Moment.txt', '-ele')
with open('Po_Displacement.txt', 'r') as file:
    node_disp_data = [line.strip().split() for line in file]
with open('Po_Moment.txt', 'r') as file:
    ele_nmvm_data = [line.strip().split() for line in file]
node_disp_df = pd.DataFrame(node_disp_data)
ele_nmvm_df = pd.DataFrame(ele_nmvm_data)
with pd.ExcelWriter('Esforços Equivalentes PO c2.xlsx') as writer:
    node_disp_df.to_excel(writer, sheet_name='Po_Displacement', index=False,
header=False)
    ele_nmvm_df.to_excel(writer, sheet_name='Po_Moment', index=False,
header=False)
wb_Po_c2 = load_workbook(r"C:/Users/luton/11. TendonTech 1.0 (BETA)/Esforços
Equivalentes PO c2.xlsx", data_only=True)
Po_NMVM_c2 = wb_Po_c2['Po_NMVM_c2']
valores_coluna_I = []
valor_inicial = Po_NMVM_c2['I7'].value
if valor_inicial is not None:
    valores_coluna_I.append(valor_inicial)
indice = 8
while True:
    celula = f'I{indice}'
    valor = Po_NMVM_c2[celula].value
    if valor is None:
        break
    valores_coluna_I.append(valor)
    indice += 8
Mp_c2 = valores_coluna_I

```

O código anterior gera o modelo estático com as forças equivalentes ao pré-esforço e o respetivo diagrama de momentos, ambos apresentados na Figura 3.23.

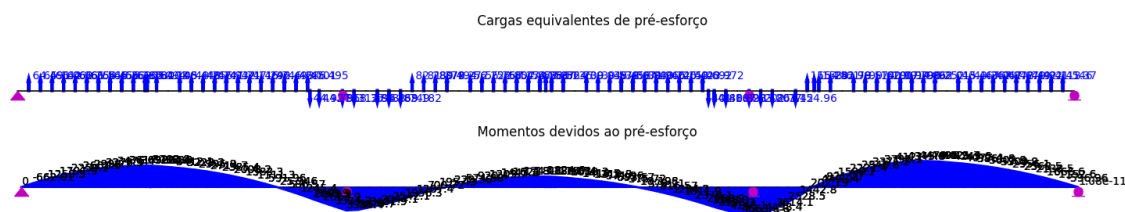


Figura 3.23 - Cargas equivalentes ao pré-esforço; Diagrama de momentos fletores associados.

Com as cargas equivalentes determinadas, calcula-se as tensões no betão resultantes dessas cargas, estimando as perdas por deformação elástica do betão, de acordo com as equações (12) e (13). O bloco de código correspondente é apresentado na Tabela 3.20.

Tabela 3.20 - Código necessário para calcular as perdas devidas ao encurtamento elástico do betão.

```

if n_cabos == 2:
    print('\nCABO II: Pré-esforço após perdas imediatas:')
    data_Pmo_c2 = {
        'Secção': perdas_sl_c2['Secção'],
        'x [m]': perdas_sl_c2['x [m]'],
        'Pmo [kN]': perdas_sl_c2['Psl [kN]']
    }
    if Pmax == Pmax_vao:
        data_Pmo_c2['S_cg [MPa]'] = [round((o / Ac + (float(M)) / Iz * vi) * 10**(-3), 2)
for M in M_CP_vao]
        data_Pmo_c2['S_cp [MPa]'] = [round((-P / Ac + (float(Mp)) / Iz * vi) * 10**(-3),
2) for P, M, Mp in zip(data_Pmo_c2['Pmo [kN]'], M_CP_vao, Mp_c2)]
    elif Pmax == Pmax_apoio:
        data_Pmo_c2['S_cg [MPa]'] = [round((o / Ac + (float(M)) / Iz * vi) * 10**(-3), 2)
for M in M_CP_apoio]
        data_Pmo_c2['S_cp [MPa]'] = [round((-P / Ac + (float(Mp)) / Iz * vi) * 10**(-3),
2) for P, M, Mp in zip(data_Pmo_c2['Pmo [kN]'], M_CP_apoio, Mp_c2)]
        data_Pmo_c2['eps_c [mm/m]'] = [round((S_cg + S_cp) / (Ecm * 10**(-6)), 2) for
S_cg, S_cp in zip(data_Pmo_c2['S_cg [MPa]'], data_Pmo_c2['S_cp [MPa]'])]
        data_Pmo_c2['d_sl [%]'] = perdas_sl_c2['d_sl [%]']
        Pmo_C2 = pd.DataFrame(data_Pmo_c2)
        print(Pmo_C2)
    print('\nCABO I: Pré-esforço após perdas por deformação elástica do betão:')

```

```

data_el_c1 = {
    'Secção': perdas_sl_c1['Secção'],
    'x [m]': perdas_sl_c1['x [m]'],
    'Psl [kN]': perdas_sl_c1['Psl [kN]'],
    }
if Pmax == Pmax_vao:
    data_el_c1['S_cg [MPa]'] = [round((o / Ac + (float(M)) / Iz * vi) * 10**(-3), 2) for
M in M_CP_vao]
    elif Pmax == Pmax_apoio:
        data_el_c1['S_cg [MPa]'] = [round((o / Ac + (float(M)) / Iz * vi) * 10**(-3), 2) for
M in M_CP_apoio]
        data_el_c1['S_cp [MPa]'] = [round((-P / Ac + (float(Mp)) / Iz * vi) * 10**(-3), 2) for
P, Mp in zip(perdas_sl_c2['Psl [kN]'], Mp_c2)]
        data_el_c1['eps_c [mm/m]'] = [round((S_cg + S_cp) / (Ecm * 10**(-6)), 2) for
S_cg, S_cp in zip(data_el_c1['S_cg [MPa]'], data_el_c1['S_cp [MPa]'])]
        data_el_c1['dPel [kN]'] = [round(abs(Ep * eps_c * 10**(-3) * Ap)) for eps_c in
data_el_c1['eps_c [mm/m]']]
        data_el_c1['Pel [kN]'] = [round(Psl - dPel) for Psl, dPel in zip(data_el_c1['Psl
[kN]'], data_el_c1['dPel [kN]'])]
        data_el_c1['d_el [%]'] = [round(d_sl + (dPel / Pmax) * 100, 2) for d_sl, dPel in
zip(data_sl_sl2_c1['d_sl [%]'], data_el_c1['dPel [kN]'])]
        perdas_el_c1 = pd.DataFrame(data_el_c1)
        print(perdas_el_c1)
        plt.figure(figsize=(20, 5))
        plt.plot(data_el_c1['x [m]'], data_el_c1['Pel [kN]'], marker='o', linestyle='-',
label='CABO I:  $\Delta P_{el}$ ')
        plt.title('CABO I: Pré-esforço após perdas por deformação elástica do betão devidas
ao esticamento do cabo 2,  $\Delta P_{el}$ ')
        plt.xlabel('Comprimento da viga, L [m]')
        plt.ylabel('$P_{el}$ [kN]')
        plt.legend()
        plt.grid(True)
        plt.show()
        print('\nCABO I: Pré-esforço após perdas imediatas:')
data_Pmo_c1 = {
    'Secção': data_el_c1['Secção'],
    'x [m]': data_el_c1['x [m]'],
    'Pmo [kN]': data_el_c1['Pel [kN]'],
    }
data_Pmo_c1['S_cg [MPa]'] = data_el_c1['S_cg [MPa]']
data_Pmo_c1['S_cp [MPa]'] = data_el_c1['S_cp [MPa]']

```

```

data_Pmo_c1['eps_c [mm/m]'] = data_el_c1['eps_c [mm/m]']
data_Pmo_c1['d_el [%]'] = data_el_c1['d_el [%]']
Pmo_c1 = pd.DataFrame(data_Pmo_c1)
print(Pmo_c1)
else:
    pass
else:
    print('\nAtenção, verificar o número de cabos!\n')
    sys.exit()
perdas_imediatas(n_estic, n_cabos)

```

As perdas devidas à deformação elástica do betão para o cabo 1, causadas pelo tensionamento do cabo 2, estão representadas na Figura 3.24. O gráfico correspondente é apresentado na Figura 3.25.

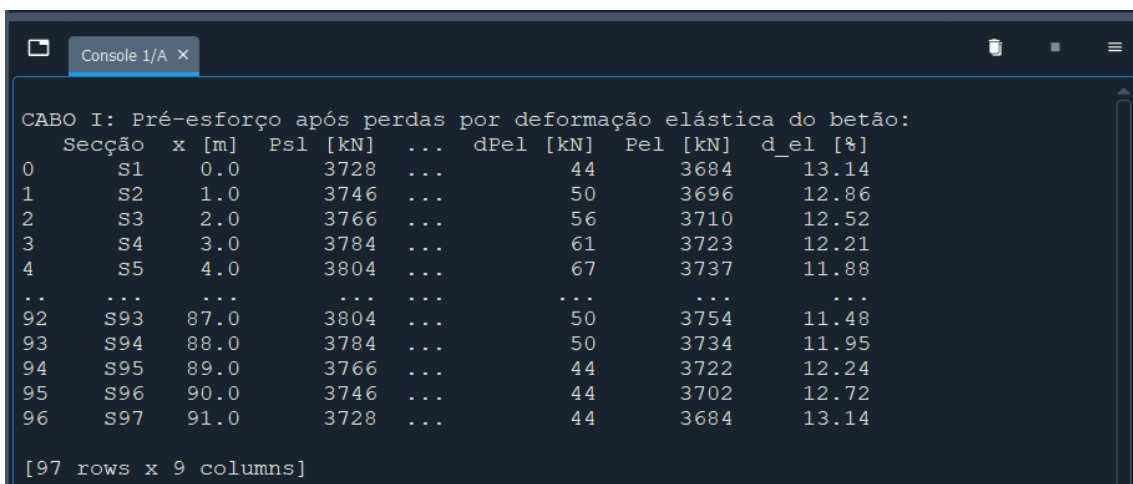


Figura 3.24 - Perdas devidas à deformação elástica do betão.

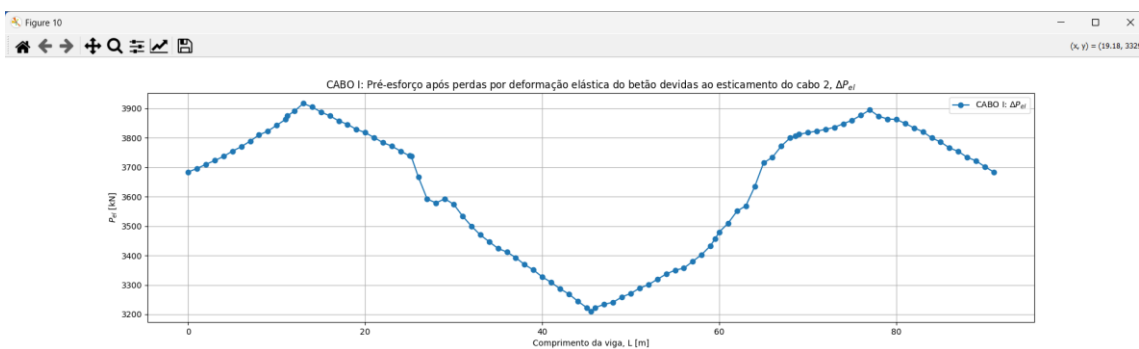


Figura 3.25 - Variação das perdas devidas à deformação elástica do betão.

3.4.5. Quantificação das perdas diferidas

As perdas diferidas são calculadas de acordo com a equação (14), que considera os efeitos diferidos no tempo do betão e a relaxação das armaduras ativas, ajustados por um fator de correção.

O coeficiente de fluência foi determinado em conformidade com o disposto no Anexo B da NP EN 1992-1-1 [5]. O bloco de código utilizado para calcular este coeficiente está apresentado na Tabela 3.21.

Tabela 3.21 – Código necessário para determinar o coeficiente de fluência.

```
def fluencia_betao(fcm, Ac, u, RH, t, to):
    global ho, phi_t_to
    a1 = (35 / fcm)**0.7
    a2 = (35 / fcm)**0.2
    a3 = (35 / fcm)**0.5
    ho = 2 * Ac / u * 10**3
    if fcm <= 35:
        B_h = 1.5 * (1 + (0.012 * RH)**18) * ho + 250
        if B_h > 1500:
            B_h = 1500
    else:
        B_h = 1.5 * (1 + (0.012 * RH)**18) * ho + 250 * a3
        if B_h > 1500 * a3:
            B_h = 1500 * a3
    B_c_t_to = ((t - to) / (B_h + t - to))**(0.3) # [-]
    B_to = 1 / (0.1 + to**(0.2))
    B_fcm = 16.8 / np.sqrt(fcm)
    if fcm <= 35:
        phi_RH = 1 + (1 - RH / 100) / (0.1 * np.cbrt(ho))
    else:
        phi_RH = (1 + (1 - RH / 100) / (0.1 * np.cbrt(ho))) * a1 * a2
    phi_o = phi_RH * B_fcm * B_to
    phi_t_to = phi_o * B_c_t_to # [-]
    fluencia_betao(fcm, Ac, u, RH, t, to)
```

A extensão de retração foi determinada como a soma da retração por secagem e da retração autogénea, em conformidade com a NP EN 1992-1-1 [5]. O bloco de código utilizado para este cálculo encontra-se apresentado na Tabela 3.22.

Tabela 3.22 – Código necessário para determinar a extensão de retração.

```
def retracao_betao(c_cim, RH, RHO, fcm, fcmo, t, to, ho, fck):
```

```

global eps_cd, eps_ca, eps_cs
if not c_cim:
    a_ds1 = 4.
elif c_cim in ['S', 's']:
    a_ds1 = 3.
elif c_cim in ['N', 'n']:
    a_ds1 = 4.
elif c_cim in ['R', 'r']:
    a_ds1 = 6.
else:
    a_ds1 = 4.
if not c_cim:
    a_ds2 = 0.12
elif c_cim in ['S', 's']:
    a_ds2 = 0.13
elif c_cim in ['N', 'n']:
    a_ds2 = 0.12
elif c_cim in ['R', 'r']:
    a_ds2 = 0.11
else:
    a_ds2 = 0.12
B_RH = 1.55 * (1 - (RH / RHo)**3)
eps_cdo = 0.85 * ((220 + 110 * a_ds1) * math.exp(-a_ds2 * fcm / fcmo)) * 10**(-6) * B_RH
# [mm/m]
Bds_t_ts = (t - to) / (t - to + 0.04 * np.sqrt(ho**3))
dados_kh = {
    100: 1,
    200: 0.85,
    300: 0.75,
    500: 0.7
}
if ho >= 500:
    kh = 0.7
elif ho < 100:
    kh = 1
else:
    ho_anterior = max([h for h in dados_kh if h <= ho])
    ho_proximo = min([h for h in dados_kh if h >= ho])
    kh_anterior = dados_kh[ho_anterior]
    kh_proximo = dados_kh[ho_proximo]

```

```

    kh = kh_anterior + ((kh_proximo - kh_anterior) / (ho_proximo - ho_anterior)) * (ho -
ho_anterior)
    eps_cd = Bds_t_ts * kh * eps_cdo # Retração por secagem [mm/m]
    B_as = 1 - math.exp(-0.2 * t**0.5)
    eps_ca_oo = 2.5 * (fck - 10) * 10**(-6)
    eps_ca = B_as * eps_ca_oo # Retração Autogénea [mm/m]
    eps_cs = eps_cd + eps_ca # Retração total, [mm/m]
retracao_betao(c_cim, RH, RHO, fcm, fcmo, t, to, ho, fck)

```

Para a relaxação do aço, também em conformidade com a NP EN 1992-1-1 [5], tem em conta três classes de relaxação, sendo a Classe 1, relaxação normal, Classe 2, baixa relaxação e a Classe 3 para barras laminadas a quente e com tratamento complementar. Apresenta-se na Tabela 3.23 o excerto de código necessário para calcular os seus valores.

Tabela 3.23 – Código necessário para determinar a relaxação do aço.

```

def relaxacao_aco(n_cabos):
    if not n_cabos or n_cabos > 10 or n_cabos <= 0:
        print("\nNúmero de cabos não suportado na presente versão!\n(*** TendonTech 1.0
2024 ***)')
        sys.exit()
    else:
        if n_cabos >= 1:
            global rho_1000, S_pr_c1
            if n_cabos == 1:
                S_pi_c1 = [(Pmo * 10**(-3)) / Ap for Pmo in Pmo_C1['Pmo [kN]']]
            else:
                S_pi_c1 = [(Pmo * 10**(-3)) / Ap for Pmo in Pmo_c1['Pmo [kN]']]
            mu_pi_c1 = [S_pi / (fpk * 10**(-3)) for S_pi in S_pi_c1]
            if not c_aco_PE:
                rho_1000 = 2.5 # [%]
            elif c_aco_PE == 1:
                rho_1000 = 8. # [%]
            elif c_aco_PE == 2:
                rho_1000 = 2.5 # [%]
            elif c_aco_PE == 3:
                rho_1000 = 4. # [%]
            else:
                rho_1000 = 2.5 # [%]
            if not c_aco_PE:

```

```

    S_pr_c1 = [0.66 * rho_1000 * math.exp(9.1 * mu_pi) * (t_pi / 1000)**(0.75 * (1 -
mu_pi)) * 10**(-5) * S_pi for mu_pi, S_pi in zip(mu_pi_c1, S_pi_c1)]
    elif c_aco_PE == 1:
        S_pr_c1 = [5.39 * rho_1000 * math.exp(6.7 * mu_pi) * (t_pi / 1000)**(0.75 * (1 -
mu_pi)) * 10**(-5) * S_pi for mu_pi, S_pi in zip(mu_pi_c1, S_pi_c1)]
    elif c_aco_PE == 2:
        S_pr_c1 = [0.66 * rho_1000 * math.exp(9.1 * mu_pi) * (t_pi / 1000)**(0.75 * (1 -
mu_pi)) * 10**(-5) * S_pi for mu_pi, S_pi in zip(mu_pi_c1, S_pi_c1)]
    elif c_aco_PE == 3:
        S_pr_c1 = [1.98 * rho_1000 * math.exp(8. * mu_pi) * (t_pi / 1000)**(0.75 * (1 -
mu_pi)) * 10**(-5) * S_pi for mu_pi, S_pi in zip(mu_pi_c1, S_pi_c1)]
    else:
        S_pr_c1 = [0.66 * rho_1000 * math.exp(9.1 * mu_pi) * (t_pi / 1000)**(0.75 * (1 -
mu_pi)) * 10**(-5) * S_pi for mu_pi, S_pi in zip(mu_pi_c1, S_pi_c1)]
    relaxacao_aco(n_cabos)

```

Com os efeitos diferidos no tempo do betão e a relaxação do aço determinados, procede-se ao cálculo das perdas diferidas. O respetivo bloco de código encontra-se apresentado na Tabela 3.24.

Tabela 3.24 – Código necessário para o cálculo das perdas diferidas.

```

def perdas_diferidas(n_cabos):
    if not n_cabos or n_cabos > 10 or n_cabos <= 0:
        print('\nNúmero de cabos não suportado na presente versão!\n(*** TendonTech 1.0
2024 ***)')
        sys.exit()
    else:
        if n_cabos >= 2:
            global P_util_c2
            def calcular_distancias_c2(a, c, H):
                x_values = np.linspace(0, H, num=int(H)+1)
                return [a * x**2 + c for x in x_values]
            def correcao_distancias_c2(distancia_anterior, novas_distancias):
                return [distancia_anterior + distancia for distancia in novas_distancias]
            distancia1_c2 = calcular_distancias_c2(a1_c2, c1_c2, H1)
            distancia2_c2 = correcao_distancias_c2(distancia1_c2[-1],
calcular_distancias_c2(a2_c2, o., H2))
            distancia3_c2 = correcao_distancias_c2(distancia2_c2[-1],
calcular_distancias_c2(a3_c2, o., H3))
            distancia4_c2 = correcao_distancias_c2(distancia3_c2[-1],
calcular_distancias_c2(a4_c2, o., H4))

```

```

        distancia5_c2 = correcao_distancias_c2(distancia4_c2[-1],
calcular_distancias_c2(a5_c2, o., H5))
        distancia6_c2 = correcao_distancias_c2(distancia5_c2[-1],
calcular_distancias_c2(a6_c2, o., H6))
        distancia7_c2 = correcao_distancias_c2(distancia6_c2[-1],
calcular_distancias_c2(a7_c2, o., H7))
        distancia8_c2 = correcao_distancias_c2(distancia7_c2[-1],
calcular_distancias_c2(a8_c2, o., H8))
        distancia9_c2 = correcao_distancias_c2(distancia8_c2[-1],
calcular_distancias_c2(a9_c2, o., H9))
        distancia10_c2 = correcao_distancias_c2(distancia9_c2[-1],
calcular_distancias_c2(a10_c2, c10_c2, H10))

        Zcp_c2 = distancia1_c2 + distancia2_c2 + distancia3_c2 + distancia4_c2 +
distancia5_c2 + distancia6_c2 + distancia7_c2 + distancia8_c2 + distancia9_c2 +
distancia10_c2
        if len(Zcp_c2) > len(nos):
            Zcp_c2 = Zcp_c2[:len(nos)]
        elif len(Zcp_c2) < len(nos):
            Zcp_c2.extend([Zcp_c2[-1]] * (len(nos) - len(Zcp_c2)))
        if n_cabos == 2:
            dPdif_c2 = [round(Ap * (eps_cs * Ep + 0.8 * S_pr + Ep / Ecm * phi_t_to * (S_cg +
S_cp)) / (1 + (Ep / Ecm) * (Ap / Ac) * (1 + Ac / Iz * Zcp**2) * (1 + 0.8 * phi_t_to)))
            for S_pr, S_cg, S_cp, Zcp in zip(S_pr_c2, Pmo_C2['S_cg [MPa]'],
Pmo_C2['S_cp [MPa]'], Zcp_c2)]
        else:
            dPdif_c2 = [round(Ap * (eps_cs * Ep + 0.8 * S_pr + Ep / Ecm * phi_t_to * (S_cg +
S_cp)) / (1 + (Ep / Ecm) * (Ap / Ac) * (1 + Ac / Iz * Zcp**2) * (1 + 0.8 * phi_t_to)))
            for S_pr, S_cg, S_cp, Zcp in zip(S_pr_c2, Pmo_c2['S_cg [MPa]'],
Pmo_c2['S_cp [MPa]'], Zcp_c2)]
        print('\nCABO II: Pré-esforço após perdas diferidas:')
        if n_cabos == 2:
            data_dif_c2 = {
                'Secção': Pmo_C2['Secção'],
                'x [m]': Pmo_C2['x [m]'],
                'Pmo [kN]': Pmo_C2['Pmo [kN]'],
            }
            data_dif_c2['dPdif [kN]'] = dPdif_c2
            data_dif_c2['P_util [kN]'] = [round(Pel - dPdif) for Pel, dPdif in zip(Pmo_C2['Pmo
[kN]'], data_dif_c2['dPdif [kN]'])]
            data_dif_c2['d_dif [%]'] = [round(d_sl + (dPdif / Pmax) * 100, 2) for d_sl, dPdif in
zip(Pmo_C2['d_sl [%]'], data_dif_c2['dPdif [kN]'])]

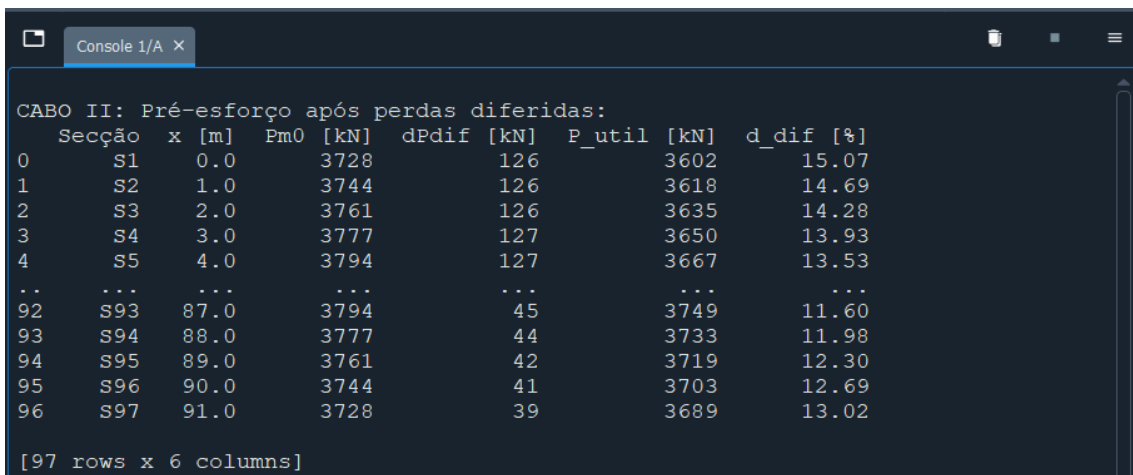
```

```

P_util_c2 = pd.DataFrame(data_dif_c2)
print(P_util_c2)
else:
    data_dif_c2 = {
        'Secção': Pmo_c2['Secção'],
        'x [m]': Pmo_c2['x [m]'],
        'Pmo [kN]': Pmo_c2['Pmo [kN]'],
    }
    data_dif_c2['dPdif [kN]'] = dPdif_c2
    data_dif_c2['P_util [kN]'] = [round(Pel - dPdif) for Pel, dPdif in zip(Pmo_c2['Pmo
[kN]'], data_dif_c2['dPdif [kN]'])]
    data_dif_c2['d_dif [%]'] = [round(d_el + (dPdif / Pmax) * 100, 2) for d_el, dPdif in
zip(Pmo_c2['d_el [%]'], data_dif_c2['dPdif [kN]'])]
    P_util_c2 = pd.DataFrame(data_dif_c2)
    print(P_util_c2)
plt.figure(figsize=(20, 5))
plt.plot(data_dif_c2['x [m]'], data_dif_c2['P_util [kN]'], marker='o', linestyle='-',
label='CABO II: Pré-esforço útil,  $\Delta P_{\text{oo}}$  ($)')
plt.title('CABO II: Pré-esforço após perdas diferidas,  $\Delta P_{\text{oo}}$  ($)')
plt.xlabel('Comprimento da viga, L [m]')
plt.ylabel('$P_{\text{oo}}$ [kN]')
plt.legend()
plt.grid(True)
plt.show()
perdas_diferidas(n_cabos)

```

O código gera uma tabela com os valores das perdas, apresentada na Figura 3.26, e um gráfico correspondente, ilustrado na Figura 3.27.



```

CABO II: Pré-esforço após perdas diferidas:
  Secção  x [m]  Pm0 [kN]  dPdif [kN]  P_util [kN]  d_dif [%]
0     S1    0.0    3728     126     3602     15.07
1     S2    1.0    3744     126     3618     14.69
2     S3    2.0    3761     126     3635     14.28
3     S4    3.0    3777     127     3650     13.93
4     S5    4.0    3794     127     3667     13.53
..     ...    ...     ...     ...     ...
92    S93   87.0    3794      45     3749     11.60
93    S94   88.0    3777      44     3733     11.98
94    S95   89.0    3761      42     3719     12.30
95    S96   90.0    3744      41     3703     12.69
96    S97   91.0    3728      39     3689     13.02

[97 rows x 6 columns]

```

Figura 3.26 - Pré-esforço após as perdas diferidas.

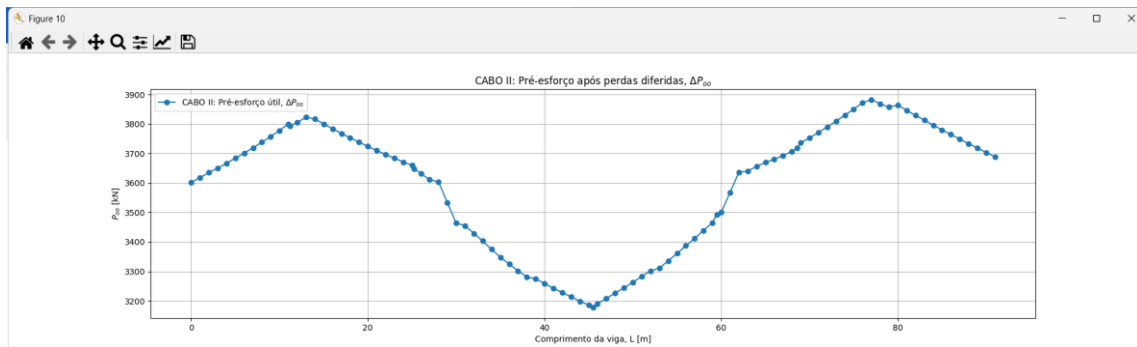


Figura 3.27 - Variação do pré-esforço após perdas diferidas.

Seguidamente, é gerada uma representação gráfica da variação total do pré-esforço utilizando o bloco de código apresentado na Tabela 3.25. O gráfico resultante encontra-se ilustrado na Figura 3.28.

Tabela 3.25 - Código necessário para determinar o pré-esforço final.

```
def pre_esforco_final():
    elif n_cabos == 2:
        global P_final_2
        print('\nPré-esforço final:')
        data_final_2 = {
            'Secção': Pmo_C2['Secção'],
            'x [m]': Pmo_C2['x [m]'],
            'Pmo [kN]': Pmo_c1['Pmo [kN]'] + Pmo_C2['Pmo [kN]'],
            'P_util [kN]': P_util_c1['P_util [kN]'] + P_util_c2['P_util [kN]']
        }
        P_final_2 = pd.DataFrame(data_final_2)
        print(P_final_2)
        plt.figure(figsize=(20, 5))
        plt.plot(data_final_2['x [m]'], data_final_2['Pmo [kN]'], marker='o', linestyle='-',
label='Pré-esforço inicial, $ΔP_{mo}$')
        plt.plot(data_final_2['x [m]'], data_final_2['P_util [kN]'], marker='o', linestyle='-',
label='Pré-esforço útil, $ΔP_{oo}$')
        plt.title('Variação do pré-esforço')
        plt.xlabel('Comprimento da viga, L [m]')
        plt.ylabel('P [kN]')
        plt.legend()
        plt.grid(True)
        plt.savefig('pfinal_dois_cabos.png')
        image = Image.open('pfinal_dois_cabos.png')
```

```

rotated_image = image.rotate(90, expand=True)
rotated_image.save('pfinal_dois_cabos_ROT.png')
plt.show()
else:
    print("\n(Pré-esforço final)\nCombinação inválida!\nVerificar o número de cabos.")
    return None
pre_esforco_final()

```

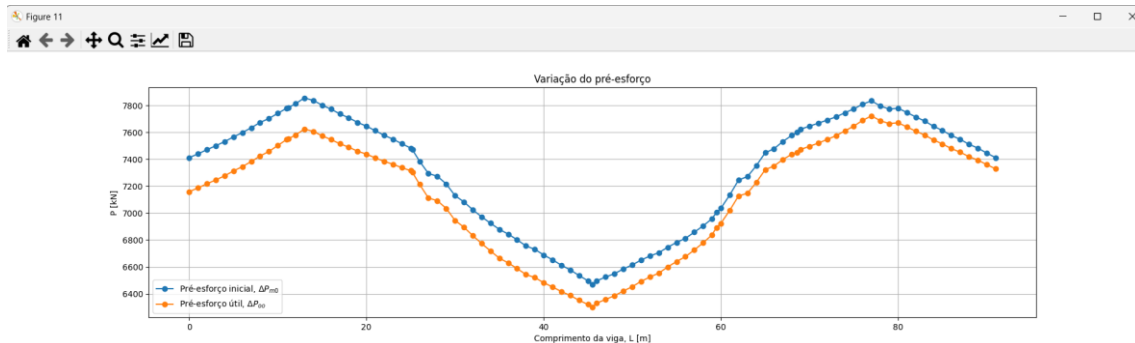


Figura 3.28 - Variação total do pré-esforço.

3.4.6. Relatório de cálculo

Após a sequência de cálculos apresentados nos blocos de código anteriores e com todos os resultados devidamente obtidos, procede-se à elaboração do relatório de cálculo automático. A Tabela 3.26 apresenta o excerto de código necessário para a criação deste relatório.

Tabela 3.26 - Código necessário para gerar o relatório de cálculo.

```

def gerar_relatorio():
    doc = Document()
    style = doc.styles['Normal']
    font = style.font
    font.name = 'Courier New'
    font.size = Pt(10)
    custom_style = doc.styles.add_style('CustomStyle', 1)
    custom_style.font.name = 'Courier New'
    custom_style.font.size = Pt(18)
    title = doc.add_heading('*** TendonTech 1.0 2024 ***', level=0)
    title.style = custom_style
    for run in title.runs:
        run.bold = True
    title.alignment = WD_PARAGRAPH_ALIGNMENT.CENTER
    now = datetime.now()

```

```

formatted_time = now.strftime("%Y-%m-%d %H:%M:%S") # Formatar como "AAAA-
MM-DD HH:MM:SS"
section = doc.sections[0]
footer = section.footer
footer_para = footer.paragraphs[0]
run = footer_para.add_run(f"TendonTech 1.0 2024 (versão BETA) - Relatório automático
- Gerado em: {formatted_time}")
run.font.size = Pt(8)
footer_para.alignment = WD_PARAGRAPH_ALIGNMENT.CENTER
doc.add_paragraph("")
doc.add_paragraph('Projeto de Aplicação de Pré-Esforço')
doc.add_paragraph(f'Viga pré-esforçada: {L1}m no vão 1, {L2}m no vão 2 e {L3}m no vão
3, com um comprimento total de {L}m.')
doc.add_paragraph("")
doc.add_paragraph("")
doc.add_picture('modelo_viga_cont3.png', width=Inches(6))
doc.add_paragraph("")
relatorio = doc.add_heading('RELATÓRIO DE CÁLCULO', 1)
relatorio.alignment = WD_PARAGRAPH_ALIGNMENT.CENTER
doc.add_paragraph("")
doc.add_heading('Convenção de sinais', level=2)
doc.add_picture('convencao_sinais.png', width=Inches(8))
doc.add_heading('Caraterísticas do Betão', level=2)
doc.add_paragraph(f'Classe: C{fck}/{{fck}}')
doc.add_paragraph(f'Módulo de elasticidade, Ecm: {{round(Ecm * 10**(-6))}} GPa')
doc.add_paragraph(f'Coeficiente de fluência,  $\phi(t,t_0)$ : {{round(phi_t_to, 2)}}')
doc.add_paragraph(f'Extensão total de retração,  $\epsilon_{cs}$ : {{round(eps_cs * 1000, 2)}} %')
doc.add_paragraph("")
doc.add_heading('Propriedades Geométricas da Secção Transversal de Betão', level=2)
doc.add_paragraph(f'Área, Ac: {{round(Ac, 3)}} m2')
doc.add_paragraph(f'Momento de inércia, Iz: {{round(Iz, 3)}} m4')
doc.add_paragraph(f'Altura, h: {{round(h, 3)}} m')
doc.add_paragraph(f'Perímetro, u: {{round(u, 3)}} m')
doc.add_paragraph(f'Distância do eixo neutro à fibra inferior, vi: {{round(vi, 3)}} m')
doc.add_paragraph(f'Distância do eixo neutro à fibra superior, vs: {{round(vs, 3)}} m')
doc.add_paragraph(f'Módulo de Resistência da fibra inferior, zi: {{round(zi, 3)}} m3')
doc.add_paragraph(f'Módulo de Resistência da fibra superior, zs: {{round(zs, 3)}} m3')
doc.add_paragraph(f'Distância do centro geométrico ao limite superior do Núcleo Central,
ki: {{round(ki, 3)}} m')
doc.add_paragraph(f'Distância do centro geométrico ao limite inferior do Núcleo Central,
ks: {{round(ks, 3)}} m')

```

```

doc.add_paragraph(f'Excentricidade limite (vãos), e_lim: {round(e_lim, 3)} m')
doc.add_paragraph(f'Excentricidade limite (apoio), e_lim: {round(e_lim_neg, 3)} m')
doc.add_paragraph("")
doc.add_heading('Caraterísticas do Sistema de Pré-Esforço', level=2)
doc.add_paragraph(f'Coeficiente de Atrito,  $\mu$ : {mu}')
doc.add_paragraph(f'Desvio Angular Parasita, k: {k} rad/m')
doc.add_paragraph(f'Reentrada das Cunhas nas ancoragem,  $\Delta s_l$ : {round(dsl * 10**3, 3)}
mm')
doc.add_paragraph("")
doc.add_heading('Caraterísticas do Aço de Pré-Esforço', level=2)
doc.add_paragraph(f'Tensão de rotura, fpuk: {round(fpk * 10**(-3))} MPa')
doc.add_paragraph(f'Módulo de elasticidade, Ep: {round(Ep * 10**(-6))} MPa')
doc.add_paragraph(f'Relaxação (1000h após aplicação de pré-esforço),  $\rho_{1000}$ :
{round(rho_1000, 3)} %')
doc.add_paragraph(f'Área nominal: {round(A_nominal * 10**4, 2)} cm2')
doc.add_paragraph("")
doc.add_heading('Caraterísticas dos cabos de Pré-esforço e Força de Tensionamento',
level=2)
if n_cabos == 1:
    table = doc.add_table(rows=3, cols=9)
elif n_cabos == 2:
    table = doc.add_table(rows=5, cols=9)
elif n_cabos == 3:
    table = doc.add_table(rows=7, cols=9)
elif n_cabos == 4:
    table = doc.add_table(rows=9, cols=9)
elif n_cabos == 5:
    table = doc.add_table(rows=11, cols=9)
elif n_cabos == 6:
    table = doc.add_table(rows=13, cols=9)
elif n_cabos == 7:
    table = doc.add_table(rows=15, cols=9)
elif n_cabos == 8:
    table = doc.add_table(rows=17, cols=9)
elif n_cabos == 9:
    table = doc.add_table(rows=19, cols=9)
elif n_cabos == 10:
    table = doc.add_table(rows=21, cols=9)
else:
    sys.exit()
hdr_cells = table.rows[0].cells

```

```

hdr_cells[0].text = 'Ordem'
hdr_cells[1].text = 'Tipo'
hdr_cells[2].text = 'Cabo'
hdr_cells[3].text = 'Extr.'
hdr_cells[4].text = 'Área [cm2]'
hdr_cells[5].text = 'L [m]'
hdr_cells[6].text = 'Bainha [mm]'
hdr_cells[7].text = 'Pmáx [kN]'
hdr_cells[8].text = 'ΔL [mm]'
elif n_cabos == 2:
    row_cells = table.rows[1].cells
    row_cells[0].text = '1o'
    row_cells[1].text = f'{{perfil_cabo['AD14'].value}}'
    row_cells[2].text = '1'
    row_cells[3].text = '1'
    row_cells[4].text = f'{{round(Ap * 10**4, 3)}}'
    def comprimento_parabolas_c1(a, x1, x2):
        integrand = lambda x: np.sqrt(1 + (2*a*x)**2)
        comprimento, _ = quad(integrand, x1, x2)
        return comprimento
    coefs_c1 = [a1_c1, a2_c1, a3_c1, a4_c1, a5_c1, a6_c1, a7_c1, a8_c1, a9_c1, a10_c1]
    comprs_c1 = [H1, H2, H3, H4, H5, H6, H7, H8, H9, H10]
    comprimentos_c1 = [comprimento_parabolas_c1(a, 0, H) for a, H in zip(coefs_c1,
comprs_c1)]
    L_cabo_c1 = sum(comprimentos_c1)
    row_cells[5].text = f'{{L_cabo_c1:.3f}}'
    row_cells[6].text = f'{{bainha}}'
    row_cells[7].text = f'{{round(Pmax)}}'
    row_cells[8].text = f'{{round(along_c1 * 10**3)}}'
    row_cells = table.rows[2].cells
    row_cells[0].text = '2o'
    row_cells[1].text = f'{{perfil_cabo['AD14'].value}}'
    row_cells[2].text = '1'
    row_cells[3].text = '2'
    row_cells[4].text = f'{{round(Ap * 10**4, 3)}}'
    row_cells[5].text = f'{{1/2 * L_cabo_c1:.3f}}'
    row_cells[6].text = f'{{bainha}}'
    row_cells[7].text = f'{{round(Pmax)}}'
    row_cells[8].text = f'{{round(along_2_c1 * 10**3)}}'
    row_cells = table.rows[3].cells
    row_cells[0].text = '3o'

```

```

row_cells[1].text = f'{perfil_cabo['AD30'].value}'
row_cells[2].text = '2'
row_cells[3].text = '1'
row_cells[4].text = f'{round(Ap * 10**4, 3)}'
def comprimento_parabolas_c2(a, x1, x2):
    integrand = lambda x: np.sqrt(1 + (2*a*x)**2)
    comprimento, _ = quad(integrand, x1, x2)
    return comprimento
coefs_c2 = [a1_c2, a2_c2, a3_c2, a4_c2, a5_c2, a6_c2, a7_c2, a8_c2, a9_c2, a10_c2]
comprs_c2 = [H1, H2, H3, H4, H5, H6, H7, H8, H9, H10]
comprimentos_c2 = [comprimento_parabolas_c2(a, 0, H) for a, H in zip(coefs_c2,
comprs_c2)]
L_cabo_c2 = sum(comprimentos_c2)
row_cells[5].text = f'{L_cabo_c2:.3f}'
row_cells[6].text = f'{bainha}'
row_cells[7].text = f'{round(Pmax)}'
row_cells[8].text = f'{round(along_c2 * 10**3)}'
row_cells = table.rows[4].cells
row_cells[0].text = '4°'
row_cells[1].text = f'{perfil_cabo['AD30'].value}'
row_cells[2].text = '2'
row_cells[3].text = '2'
row_cells[4].text = f'{round(Ap * 10**4, 3)}'
row_cells[5].text = f'{1/2 * L_cabo_c2:.3f}'
row_cells[6].text = f'{bainha}'
row_cells[7].text = f'{round(Pmax)}'
row_cells[8].text = f'{round(along_2_c2 * 10**3)}'
else:
    sys.exit()
doc.add_paragraph('Nota: O Tipo 0 refere-se a uma sequência de traçados retos; o Tipo 1
refere-se a uma sequência de traçados parabólicos.')
doc.add_paragraph("")
doc.add_heading('Cargas e Fatores de Redução', level=2)
doc.add_paragraph(f'Carga permanente, g: {g} kN/m')
doc.add_paragraph(f'Carga acidental, q: {q} kN/m')
doc.add_paragraph(f' $\psi_2$ : {psi_2}')
doc.add_paragraph("")
doc.add_heading('Perdas iniciais de Pré-Esforço', level=2)
elif n_cabos == 2:
    doc.add_heading('CABO I: Pré-Esforço após perdas por atrito,  $P\mu$ ', level=2)

```

```

table = doc.add_table(rows=perdas_mu_c1.shape[0] + 1,
cols=perdas_mu_c1.shape[1])
hdr_cells = table.rows[0].cells
for idx, column in enumerate(perdas_mu_c1.columns):
    hdr_cells[idx].text = str(column)
for i in range(perdas_mu_c1.shape[0]):
    row_cells = table.rows[i + 1].cells
    for j in range(perdas_mu_c1.shape[1]):
        row_cells[j].text = str(perdas_mu_c1.iat[i, j])
doc.add_heading('CABO I: Pré-Esforço após perdas por reentrada das cunhas, Psl',
level=2)
table = doc.add_table(rows=perdas_sl_c1.shape[0] + 1, cols=perdas_sl_c1.shape[1])
hdr_cells = table.rows[0].cells
for idx, column in enumerate(perdas_sl_c1.columns):
    hdr_cells[idx].text = str(column)
for i in range(perdas_sl_c1.shape[0]):
    row_cells = table.rows[i + 1].cells
    for j in range(perdas_sl_c1.shape[1]):
        row_cells[j].text = str(perdas_sl_c1.iat[i, j])
doc.add_heading('CABO I: Pré-Esforço após perdas por deformação elástica do betão,
Pel', level=2)
table = doc.add_table(rows=perdas_el_c1.shape[0] + 1, cols=perdas_el_c1.shape[1])
hdr_cells = table.rows[0].cells
for idx, column in enumerate(perdas_el_c1.columns):
    hdr_cells[idx].text = str(column)
for i in range(perdas_el_c1.shape[0]):
    row_cells = table.rows[i + 1].cells
    for j in range(perdas_el_c1.shape[1]):
        row_cells[j].text = str(perdas_el_c1.iat[i, j])
doc.add_paragraph('Obs: Perdas por deformação elástica do betão no cabo 1 devidas ao
esticamento do cabo 2.')
doc.add_paragraph("")
doc.add_heading('CABO II: Pré-Esforço após perdas por atrito, Pμ', level=2)
table = doc.add_table(rows=perdas_mu_c2.shape[0] + 1,
cols=perdas_mu_c2.shape[1])
hdr_cells = table.rows[0].cells
for idx, column in enumerate(perdas_mu_c2.columns):
    hdr_cells[idx].text = str(column)
for i in range(perdas_mu_c2.shape[0]):
    row_cells = table.rows[i + 1].cells
    for j in range(perdas_mu_c2.shape[1]):

```

```

        row_cells[j].text = str(perdas_mu_c2.iat[i, j])
    doc.add_heading('CABO II: Pré-Esforço após perdas por reentrada das cunhas, Psl',
level=2)
    table = doc.add_table(rows=perdas_sl_c2.shape[0] + 1, cols=perdas_sl_c2.shape[1])
    hdr_cells = table.rows[0].cells
    for idx, column in enumerate(perdas_sl_c2.columns):
        hdr_cells[idx].text = str(column)
    for i in range(perdas_sl_c2.shape[0]):
        row_cells = table.rows[i + 1].cells
        for j in range(perdas_sl_c2.shape[1]):
            row_cells[j].text = str(perdas_sl_c2.iat[i, j])
else:
    sys.exit()
doc.add_heading('Pré-Esforço após Perdas Instantâneas, Pmo', level=2)
elif n_cabos == 2:
    doc.add_paragraph('Cabo I:')
    table = doc.add_table(rows=Pmo_c1.shape[0] + 1, cols=Pmo_c1.shape[1])
    hdr_cells = table.rows[0].cells
    for idx, column in enumerate(Pmo_c1.columns):
        hdr_cells[idx].text = str(column)
    for i in range(Pmo_c1.shape[0]):
        row_cells = table.rows[i + 1].cells
        for j in range(Pmo_c1.shape[1]):
            row_cells[j].text = str(Pmo_c1.iat[i, j])
    doc.add_paragraph('Cabo II:')
    table = doc.add_table(rows=Pmo_C2.shape[0] + 1, cols=Pmo_C2.shape[1])
    hdr_cells = table.rows[0].cells
    for idx, column in enumerate(Pmo_C2.columns):
        hdr_cells[idx].text = str(column)
    for i in range(Pmo_C2.shape[0]):
        row_cells = table.rows[i + 1].cells
        for j in range(Pmo_C2.shape[1]):
            row_cells[j].text = str(Pmo_C2.iat[i, j])
else:
    sys.exit()
doc.add_heading('Pré-Esforço Final', level=2)
elif n_cabos == 2:
    table = doc.add_table(rows=P_final_2.shape[0] + 1, cols=P_final_2.shape[1])
    hdr_cells = table.rows[0].cells
    for idx, column in enumerate(P_final_2.columns):
        hdr_cells[idx].text = str(column)

```

```

for i in range(P_final_2.shape[0]):
    row_cells = table.rows[i + 1].cells
    for j in range(P_final_2.shape[1]):
        row_cells[j].text = str(P_final_2.iat[i, j])
else:
    sys.exit()
doc.add_page_break()
doc.add_heading('Anexos', level=2)
doc.add_heading('Pré-esforço útil mínimo - Vãos', level=3)
doc.add_picture('magnel_vaos.png', width=Inches(7))
doc.add_page_break()
doc.add_heading('Pré-esforço útil mínimo - Apoios', level=3)
doc.add_picture('magnel_apoios.png', width=Inches(7))
doc.add_page_break()
elif n_cabos == 2:
    doc.add_heading('Variação do pré-esforço', level=3)
    doc.add_picture('pfinal_dois_cabos_ROT.png', height=Inches(8.2))
    doc.add_page_break()
else:
    return 0
docx_file = 'TT_CalcReport.docx'
doc.save(docx_file)
pdf_file = 'TT_CalcReport.pdf'
convert(docx_file, pdf_file)
gerar_relatorio()

```

Capítulo 4

4. Aplicação prática do programa de cálculo automático

4.1. Descrição do estudo de caso

O exemplo analisado nesta secção foi desenvolvido no âmbito da unidade curricular (10304/15298) Betão Pré-Esforçado, durante o ano letivo 2023/2024. O objetivo consistiu em determinar o pré-esforço útil na secção do apoio central de uma viga contínua com dois tramos. Na Figura 4.1, é apresentada a configuração geométrica da viga.



Figura 4.1 - Viga contínua. Exemplo de aplicação prática.

No exemplo em análise, são considerados, conforme indicado no enunciado, dois cabos com 20 cordões cada, betão C30/37, aço Y1860S7 com cordões super, sendo o valor de (n) indicado na Figura 4.1, igual a 30. Para a análise, assume-se que a viga suporta exclusivamente o seu peso próprio.

4.2. Aplicação do programa

A introdução dos dados, conforme descrito nas secções anteriores, realiza-se através do preenchimento do ficheiro TT_InputData.xlsx, onde cada folha é utilizada para inserir as informações especificadas. Na Figura 4.2 é apresentado um exemplo do preenchimento inicial, correspondente ao tipo de viga em análise.

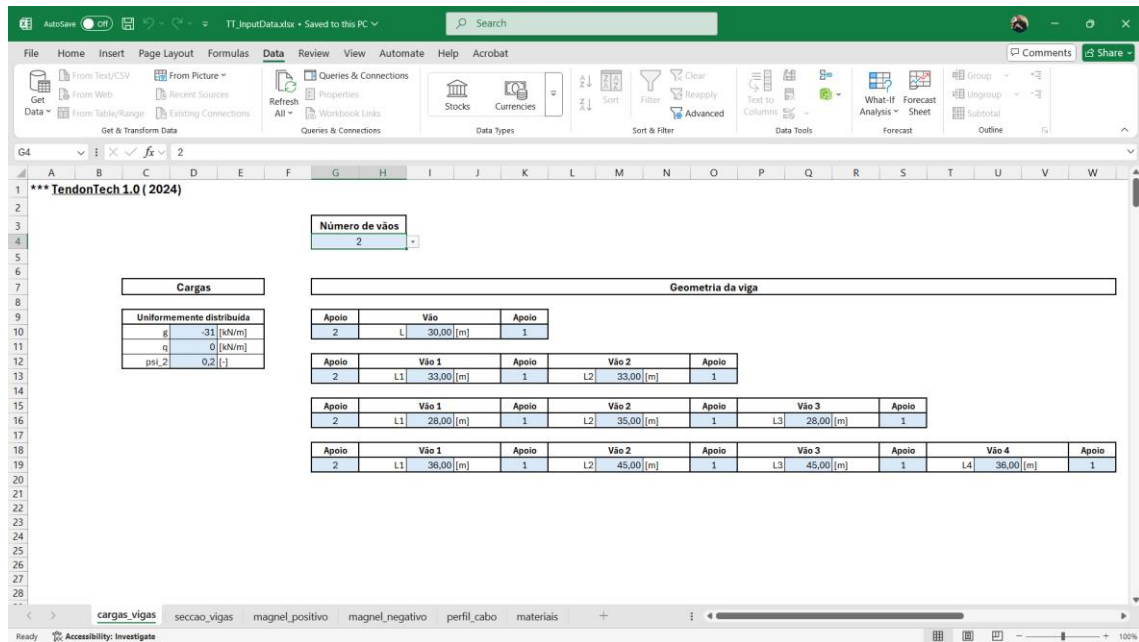


Figura 4.2 - TendonTech: Definição de cargas e características da viga. Viga contínua com dois tramos iguais de 33 m e carregamento uniformemente distribuído de 31 kN/m.

De seguida, procede-se ao preenchimento dos dados relativos à secção transversal da viga, conforme ilustrado na Figura 4.3.

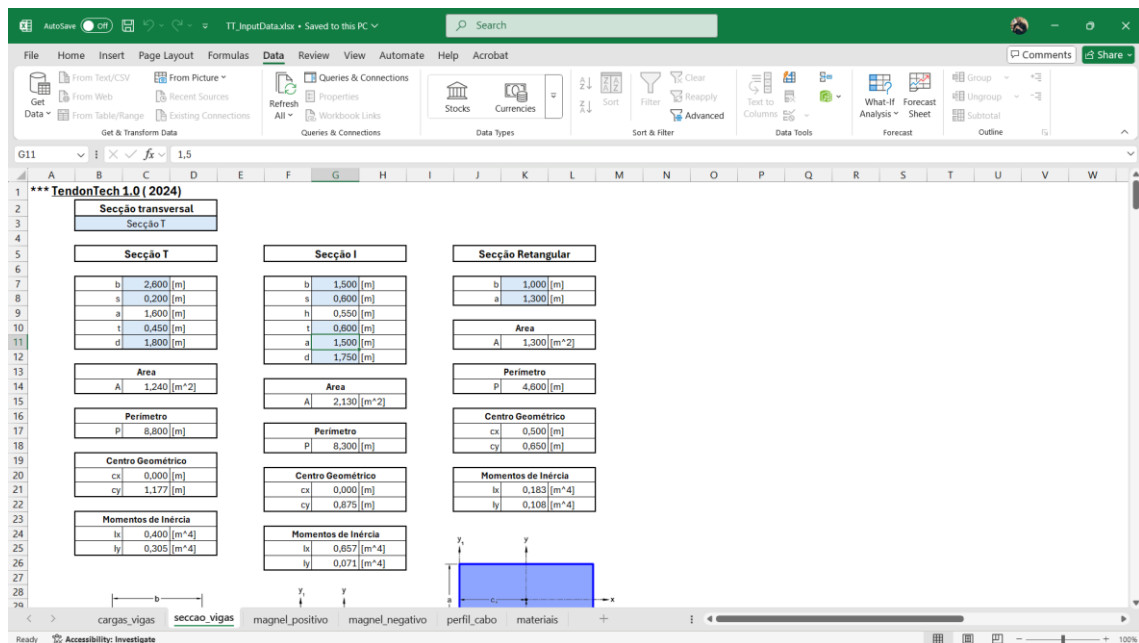


Figura 4.3 - TendonTech: Configuração da secção transversal da viga. Secção transversal em "T".

Em seguida, procede-se à definição dos materiais especificados, incluindo o aço de pré-esforço e o betão. A Figura 4.4 apresenta o preenchimento destes dados.

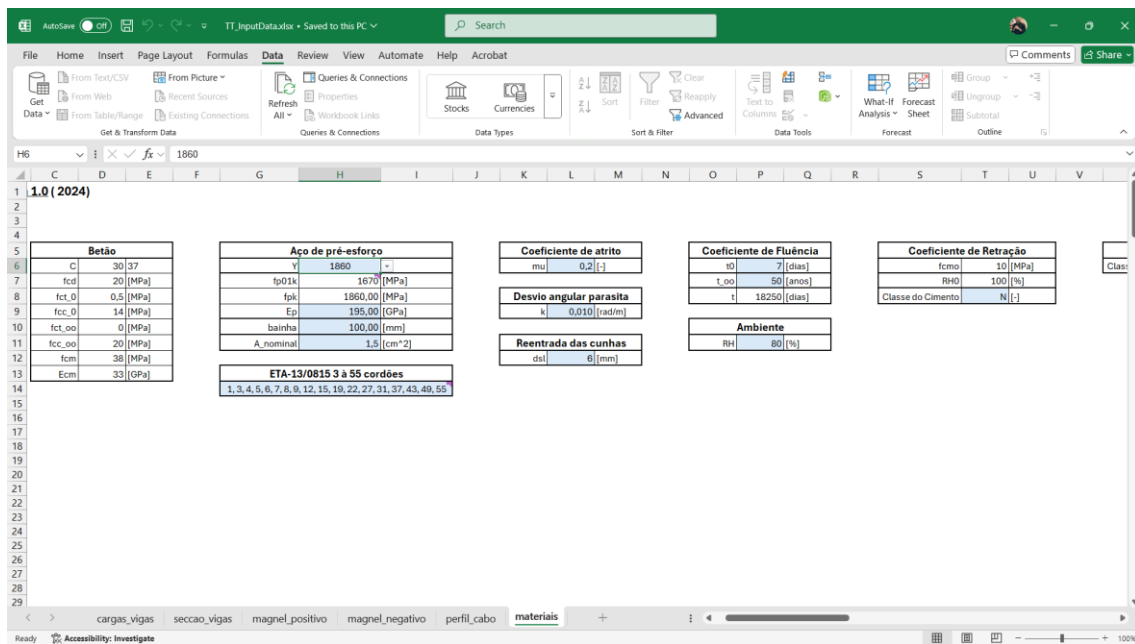


Figura 4.4 - TendonTech: Definição dos materiais. Betão C30/37 e aço de pré-esforço Y1860.

Importa referir que o enunciado especifica a utilização de 2 cabos com 20 cordões cada. No entanto, o número de cabos pré-definidos na base de dados do programa (1, 3, 4, 5, 6, 7, 8, 9, 12, 15, 19, 22, 27, 31, 37, 43, 49, 55) não inclui a opção de 20 cordões. Assim, torna-se necessário ajustar estes valores, de forma a incluir o número de cabos enunciado. Para tal, os dados serão redefinidos como (1, 3, 4, 5, 6, 7, 8, 9, 20, 27, 31, 37, 43, 49, 55), assegurando que o programa contempla 2 cabos de 20 cordões cada.

Seguidamente, procede-se ao preenchimento das informações relativas ao perfil dos cabos, conforme ilustrado na Figura 4.5.

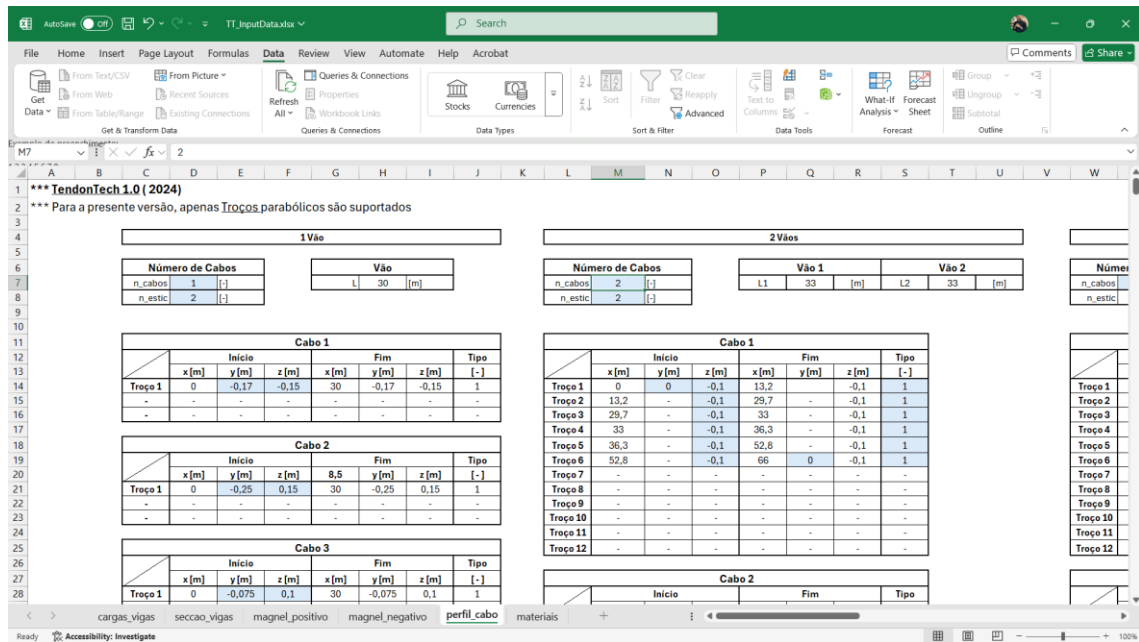


Figura 4.5 - TendonTech: Definição do perfil dos cabos. Número de cabos e número de tensionamentos especificados.

Após o preenchimento adequado do ficheiro TT_InputData.xlsx, este deve ser guardado e, em seguida, o programa deve ser executado para realizar os cálculos. O tempo de processamento dependerá da complexidade do problema em análise. Os resultados obtidos serão apresentados e discutidos na Secção 4.3 deste documento.

4.3. Análise e discussão dos resultados

A análise apresentada refere-se ao dimensionamento e comportamento de dois cabos de pré-esforçados, denominados como (CABO I e CABO II) em uma viga de betão armado. Os cabos foram modelados com um total de 20 cordões cada, apresentando uma área total de 30,0 cm² por cabo e uma força de tensionamento de 4464,0 kN, como apresenta a Figura 4.6.

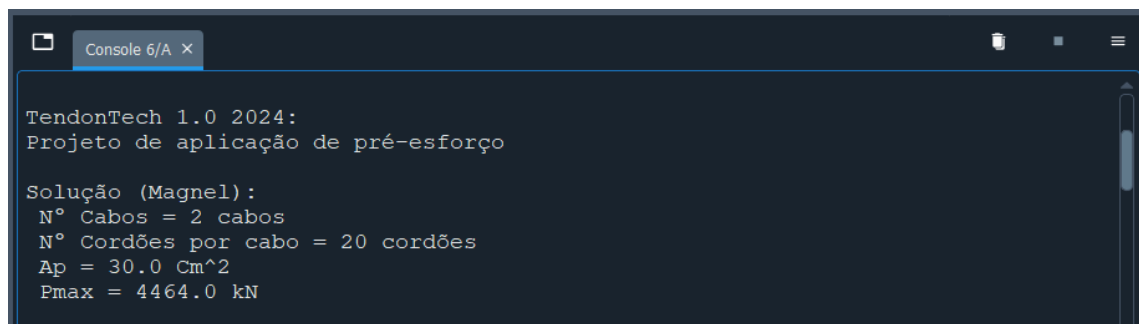


Figura 4.6 - TendonTech: Solução inicial. Número de cabos, cordões e força de tensionamento.

Os resultados incluem uma avaliação das diversas perdas, tanto imediatas como diferidas, bem como os alongamentos teóricos calculados para cada fase do tensionamento, com os valores discriminados ao longo do comprimento dos cabos.

Os resultados demonstram uma perda progressiva de força ao longo do comprimento dos cabos devido ao atrito e à curvatura do traçado. A força inicial de 4464 kN reduz-se, em média, para cerca de 4386 kN na secção intermédia S36, correspondendo a uma perda percentual acumulada de aproximadamente 14,54 %, como indicado no Anexo 1. A Figura 4.7 apresenta o resumo das perdas por atrito em ambos cabos.

Secção	x [m]	Alfa [rad]	dPmu [kN]	Pmu [kN]	d_mu [%]	
0	S1	0.0	0.0000	4464	0.00	
1	S2	1.0	0.0120	20	4444	0.45
2	S3	2.0	0.0239	39	4425	0.87
3	S4	3.0	0.0359	58	4406	1.30
4	S5	4.0	0.0479	78	4386	1.75
...	
66	S67	62.0	0.8752	78	4386	1.75
67	S68	63.0	0.8870	58	4406	1.30
68	S69	64.0	0.8987	39	4425	0.87
69	S70	65.0	0.9104	20	4444	0.45
70	S71	66.0	0.9104	0	4464	0.00

Figura 4.7 - TendonTech: Perdas por atrito nos cabos I e II.

Este padrão é simétrico nos dois cabos devido ao traçado idêntico adotado. Apresenta-se na Figura 4.8 os gráficos da perda por atrito em ambos cabos.

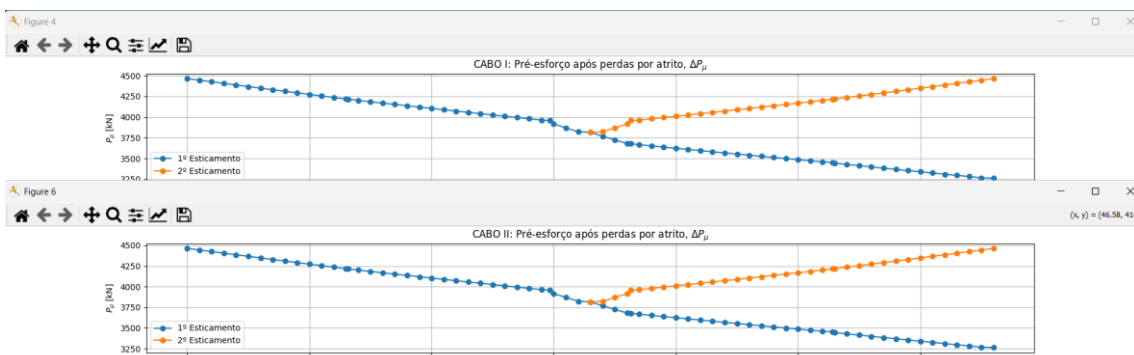


Figura 4.8 - TendonTech: Variação das perdas por atrito nos cabos I e II.

A simetria observada nos resultados valida a uniformidade no comportamento dos cabos. As perdas acumuladas, que rondam os 15 % na secção crítica do apoio interior, indicam uma otimização no traçado dos cabos. Esta configuração evita desvios excessivos e ângulos abruptos, que poderiam levar a perdas mais significativas.

As perdas associadas à reentrada das cunhas resultam em diminuições adicionais no pré-esforço dos cabos, limitadas a um determinado comprimento. Neste caso, esse comprimento, até onde a reentrada das cunhas se faz sentir, é de aproximadamente 15 m, correspondendo à secção S16. O

alongamento teórico é semelhante para ambos os cabos, no entanto, as medições no local devem ser efetuadas para validar os valores estimados, admitindo um erro prático na ordem dos 5%. Neste caso de perda, ambos os cabos demonstram um comportamento idêntico, conforme indicado nos resultados apresentados na Figura 4.9.

CABO I: Alongamento (1º esticamento) = 425.872 mm						CABO II: Alongamento (1º esticamento) = 425.872 mm					
CABO I: Perdas por reentrada das cunhas:						CABO II: Perdas por reentrada das cunhas:					
	Secção	x [m]	dPsl [kN]	Psl [kN]	d_sl [%]		Secção	x [m]	dPsl [kN]	Psl [kN]	d_sl [%]
0	S1	0.0	520	3944	11.65	0	S1	0.0	520	3944	11.65
1	S2	1.0	482	3962	11.25	1	S2	1.0	482	3962	11.25
2	S3	2.0	443	3982	10.79	2	S3	2.0	443	3982	10.79
3	S4	3.0	404	4002	10.35	3	S4	3.0	404	4002	10.35
4	S5	4.0	366	4020	9.95	4	S5	4.0	366	4020	9.95
..
66	S67	62.0	366	4020	9.95	66	S67	62.0	366	4020	9.95
67	S68	63.0	404	4002	10.35	67	S68	63.0	404	4002	10.35
68	S69	64.0	443	3982	10.79	68	S69	64.0	443	3982	10.79
69	S70	65.0	482	3962	11.25	69	S70	65.0	482	3962	11.25
70	S71	66.0	520	3944	11.65	70	S71	66.0	520	3944	11.65

[71 rows x 5 columns]

CABO I: Alongamento (2º esticamento) = 45.403 mm

CABO II: Alongamento (2º esticamento) = 45.403 mm

Figura 4.9 - TendonTech: Perdas devidas a reentrada das cunhas para os cabos I e II. Alongamentos teóricos no primeiro e segundo tensionamentos para os cabos I e II.

Os resultados apresentados na Figura 4.9 estão ilustrados de forma gráfica na Figura 4.10, mantendo-se a simetria observada. Adicionalmente, é possível identificar, nesta representação, a extensão ao longo da qual as perdas devido à reentrada das cunhas têm efeito.

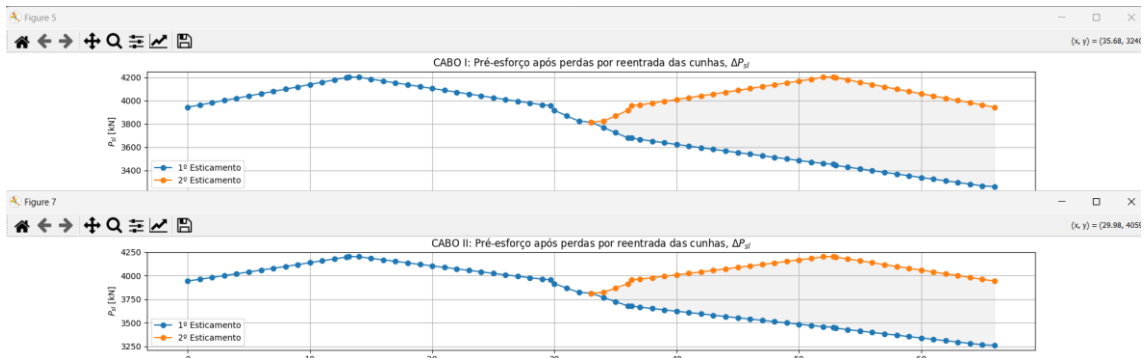


Figura 4.10 - TendonTech: Variação das perdas por reentrada das cunhas, nos cabos I e II.

Conforme referido nas secções anteriores, o tensionamento dos cabos é assumido, por defeito, como sequencial. Nesse contexto, o Cabo I apresenta perdas adicionais associadas à deformação elástica do betão resultante do tensionamento do Cabo II. Em contrapartida, o Cabo II não regista perdas associadas ao seu próprio tensionamento. A Figura 4.11 apresenta o resumo das perdas imediatas verificadas no Cabo II.

```

CABO II: Pré-esforço após perdas imediatas:
  Secção  x [m]  Pm0 [kN]  S_cg [MPa]  S_cp [MPa]  eps_c [mm/m]  d_sl [%]
0      S1    0.0    3944      -0.00      -3.18        -0.10       11.65
1      S2    1.0    3962       1.08      -4.71        -0.11       11.25
2      S3    2.0    3982       2.08      -6.10        -0.12       10.79
3      S4    3.0    4002       2.98      -7.35        -0.13       10.35
4      S5    4.0    4020       3.79      -8.46        -0.14        9.95
..     ...     ..     ...     ...     ...     ...
66     S67   62.0    4020       3.79      -9.00        -0.16        9.95
67     S68   63.0    4002       2.98      -7.72        -0.14       10.35
68     S69   64.0    3982       2.08      -6.30        -0.13       10.79
69     S70   65.0    3962       1.08      -4.74        -0.11       11.25
70     S71   66.0    3944       0.00      -3.18        -0.10       11.65

[71 rows x 7 columns]

```

Figura 4.11 - TendonTech: Pré-esforço após perdas imediatas no Cabo I. Tensões nas secções críticas ao longo do comprimento da viga. Extensões.

As perdas no Cabo I, resultantes do tensionamento do Cabo II, já não exibem uma distribuição simétrica. Contudo, os valores das perdas não se revelam significativamente expressivos, conforme os resultados apresentados na Figura 4.12.

```

CABO I: Pré-esforço após perdas por deformação elástica do betão:
  Secção  x [m]  Psl [kN]  ...  dPel [kN]  Pel [kN]  d_el [%]
0      S1    0.0    3944  ...    58    3886    12.95
1      S2    1.0    3962  ...    64    3898    12.68
2      S3    2.0    3982  ...    70    3912    12.36
3      S4    3.0    4002  ...    76    3926    12.05
4      S5    4.0    4020  ...    82    3938    11.79
..     ...     ..     ...     ...     ...     ...
66     S67   62.0    4020  ...    94    3926    12.06
67     S68   63.0    4002  ...    82    3920    12.19
68     S69   64.0    3982  ...    76    3906    12.49
69     S70   65.0    3962  ...    64    3898    12.68
70     S71   66.0    3944  ...    58    3886    12.95

[71 rows x 9 columns]

```

Figura 4.12 - TendonTech: Perdas devidas à deformação elástica do betão no Cabo I.

A representação gráfica destas perdas é apresentada na Figura 4.13.

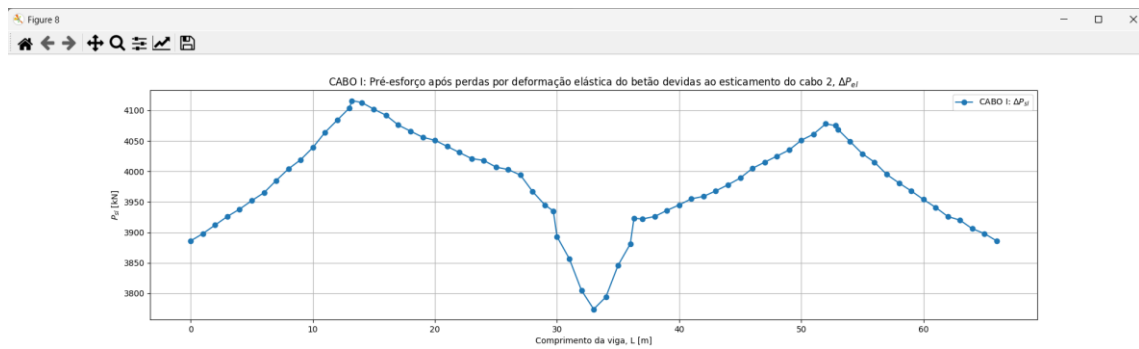


Figura 4.13 - TendonTech: Variação das perdas devidas a deformação elástica do betão no Cabo I.

O resumo das perdas no Cabo I é apresentado na Figura 4.14.

```
[71 rows x 9 columns]
```

CABO I: Pré-esforço após perdas imediatas:

	Secção	x [m]	Pm0 [kN]	S_cg [MPa]	S_cp [MPa]	eps_c [mm/m]	d_el [%]
0	S1	0.0	3886	-0.00	-3.18	-0.10	12.95
1	S2	1.0	3898	1.08	-4.71	-0.11	12.68
2	S3	2.0	3912	2.08	-6.10	-0.12	12.36
3	S4	3.0	3926	2.98	-7.35	-0.13	12.05
4	S5	4.0	3938	3.79	-8.46	-0.14	11.79
..
66	S67	62.0	3926	3.79	-9.00	-0.16	12.06
67	S68	63.0	3920	2.98	-7.72	-0.14	12.19
68	S69	64.0	3906	2.08	-6.30	-0.13	12.49
69	S70	65.0	3898	1.08	-4.74	-0.11	12.68
70	S71	66.0	3886	0.00	-3.18	-0.10	12.95

```
[71 rows x 7 columns]
```

Figura 4.14 - TendonTech: Pré-esforço após perdas imediatas no Cabo II. Tensões nas secções críticas ao longo do comprimento da viga. Extensões.

Os valores das perdas imediatas são ligeiramente superiores aos valores típicos encontrados na literatura, assim como aos 10 - 15 % estimados, por defeito, no programa.

As perdas diferidas foram calculadas em função da retração e fluência do betão e a relaxação do aço; o resumo dos valores está representado na Figura 4.15.

```

CABO I: Pré-esforço após perdas diferidas:
Secção x [m] Pm0 [kN] dPdif [kN] P_util [kN] d_dif [%]
0 S1 0.0 3886 115 3771 15.53
1 S2 1.0 3898 115 3783 15.26
2 S3 2.0 3912 115 3797 14.94
3 S4 3.0 3926 115 3811 14.63
4 S5 4.0 3938 115 3823 14.37
... ..
66 S67 62.0 3926 56 3870 13.31
67 S68 63.0 3920 56 3864 13.44
68 S69 64.0 3906 56 3850 13.74
69 S70 65.0 3898 56 3842 13.93
70 S71 66.0 3886 56 3830 14.20
[71 rows x 6 columns]

```

```

CABO II: Pré-esforço após perdas diferidas:
Secção x [m] Pm0 [kN] dPdif [kN] P_util [kN] d_dif [%]
0 S1 0.0 3944 115 3829 14.23
1 S2 1.0 3962 115 3847 13.83
2 S3 2.0 3982 115 3867 13.37
3 S4 3.0 4002 115 3887 12.93
4 S5 4.0 4020 115 3905 12.53
... ..
66 S67 62.0 4020 56 3964 11.20
67 S68 63.0 4002 56 3946 11.60
68 S69 64.0 3982 56 3926 12.04
69 S70 65.0 3962 56 3906 12.50
70 S71 66.0 3944 56 3888 12.90
[71 rows x 6 columns]

```

Figura 4.15 - TendonTech: Pré-esforço após as perdas diferidas nos cabos I e II.

Na secção crítica do apoio central, nomeadamente a secção S36, apresenta uma perda de cerca de 17 % e 16 %, para os cabos I e II, respetivamente. Esta pequena variação tem a ver com a perda adicional no Cabo I devida ao tensionamento do Cabo II. Apresenta-se na Figura 4.16, o resumo total das perdas imediatas e perdas diferidas, e na Figura 4.17, a respetiva representação gráfica.

```

[71 rows x 6 columns]
Pré-esforço final:
Secção x [m] Pm0 [kN] P_util [kN]
0 S1 0.0 7830 7600
1 S2 1.0 7860 7630
2 S3 2.0 7894 7664
3 S4 3.0 7928 7698
4 S5 4.0 7958 7728
... ..
66 S67 62.0 7946 7722
67 S68 63.0 7922 7696
68 S69 64.0 7888 7660
69 S70 65.0 7860 7632
70 S71 66.0 7830 7600
[71 rows x 4 columns]
0% | | 0/1 [00:00<?, ?it/s]

```

Figura 4.16 - TendonTech: Pré-esforço final. Pré-esforço após as perdas imediatas e pré-esforço após as perdas diferidas.

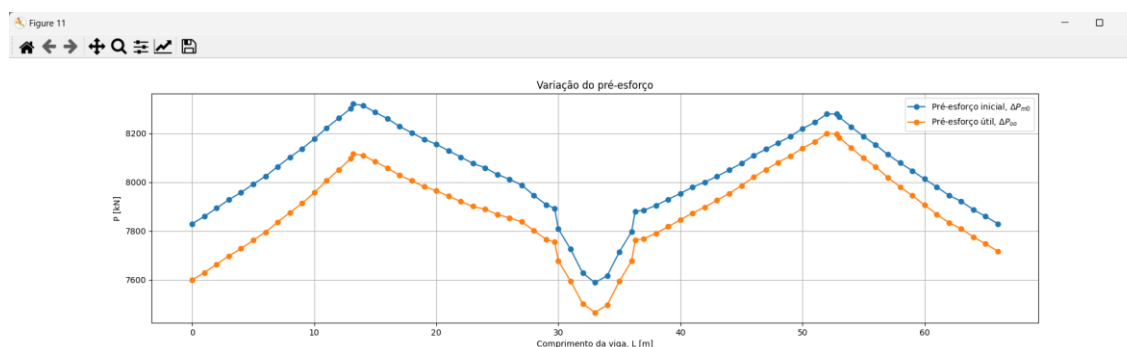


Figura 4.17 - TendonTech: Variação final do pré-esforço.

O pré-esforço útil final no cabo, em todas as secções da viga, encontra-se dentro do intervalo de 20 % estimado por defeito no programa. Na secção crítica do apoio interior, onde o enunciado solicita a determinação do pré-esforço útil, nomeadamente na secção S36, o valor obtido para P_{∞} é de 7497 kN, correspondendo a uma perda de aproximadamente 18 % em relação à força de tensionamento máxima $P_{\max} = 8928$ kN.

4.4. Validação do programa

A validação dos resultados obtidos pelo programa TendonTech é realizada por meio da comparação com os cálculos fornecidos pelo software ESTICA 2020 [6], utilizando como base o mesmo exemplo abordado e discutido no Capítulo 4, com o objetivo de avaliar a precisão dos cálculos e o comportamento estrutural das vigas de betão pré-esforçado. Este processo visa garantir a fiabilidade e a consistência do TendonTech em relação a um programa mais complexo e já utilizado no dimensionamento de estruturas reais, utilizando critérios técnicos rigorosos para validar as soluções de tensões, deformações e perdas de pré-esforço, bem como o desempenho geral dos cabos modelados. A análise comparativa permitirá identificar possíveis divergências e confirmar a capacidade do TendonTech em reproduzir resultados fiáveis.

A análise dos parâmetros da Tabela 4.1 evidencia diferenças significativas entre os resultados do ESTICA 2020 [6] e do TendonTech, sublinhando a necessidade de validar os cálculos e garantir que o TendonTech cumpre os critérios técnicos estabelecidos. A diferença de 20 % na extensão de retração poderá resultar da definição da humidade relativa (RH) e/ou da idade do betão no momento do carregamento (t_0). Além disso, a ausência de parâmetros como o comprimento do cabo no macaco e a reentrada dos cabos no mesmo, revela limitações na modelação dos efeitos locais nos dispositivos de ancoragem pelo TendonTech.

Tabela 4.1 – Validação dos resultados: Parâmetros calculados

Parâmetro	ESTICA 2020	TendonTech	Diferença
Extensão de retração, ϵ_{cs}	0,25 ‰	0,20 ‰	20,00 %
Reentrada dos cabos no macaco	1,0 mm	N/A	-
Comprimento de cabo dentro do macaco	1,0 m	N/A	-

Os resultados apresentados na Tabela 4.2 são da mesma ordem de grandeza. Isso sugere que ambos os programas seguem abordagens similares para a estimativa dos alongamentos. Essas diferenças podem estar associadas a fatores como critérios de rigidez adotados, hipóteses sobre perdas de pré-esforço ou precisão numérica dos métodos implementados.

Tabela 4.2 – Validação dos resultados: alongamentos teóricos.

Parâmetro	ESTICA 2020	TendonTech	Diferença
Alongamento teórico 1, cabo I	476 mm	426 mm	10,50 %
Alongamento teórico 2, cabo I	49 mm	45 mm	8,26 %
Alongamento teórico 1, cabo II	476 mm	426 mm	10,50 %
Alongamento teórico 2, cabo II	49 mm	45 mm	8,26 %

A Tabela 4.3 apresenta os valores de pré-esforço após as perdas imediatas e o pré-esforço útil para ambos os programas. Embora a concordância nas perdas imediatas seja satisfatória, o pré-esforço útil apresenta diferenças mais significativas, sugerindo lacunas no modelo de perdas a longo prazo do TendonTech. Esta análise evidencia a necessidade de ajustes no cálculo do pré-esforço útil para garantir resultados mais fiáveis.

Tabela 4.3 - Validação dos resultados: Pré-esforço após as perdas imediatas e pré-esforço útil.

Parâmetro		ESTICA 2020	TendonTech	Diferença
P _{max}		8928 kN		-
P _{mo}	0m	7969 kN	7830 kN	1,74 %
	17m	8268 kN	8228 kN	0,60 %
	33m	7898 kN	7572 kN	4,12 %
	50m	8416 kN	8219 kN	2,12 %
	66m	8007 kN	7830 kN	2,21 %
P _∞	0m	6906 kN	7600 kN	10,05 %
	17m	6678 kN	8022 kN	20,12 %
	33m	6524 kN	7344 kN	14,11 %
	50m	6738 kN	7991 kN	18,60 %
	66m	6941 kN	7600 kN	9,50 %

A Tabela 4.4 apresenta as tensões nas secções críticas devido ao peso próprio da viga e ao pré-esforço. Apesar das diferenças significativas nos valores, os resultados exibem um comportamento semelhante, com tensões de tração resultantes do peso próprio da viga e tensões de compressão induzidas pelo pré-esforço.

Tabela 4.4 - Validação dos resultados: tensões no betão devidas ao peso próprio e ao pré-esforço.

Parâmetro		ESTICA 2020	TendonTech	Diferença
Cabo I, σ_{cg}	0 m	0,00 MPa	0,00 MPa	-
	17 m	4,83 MPa	6,01 MPa	19,63 %
	33 m	4,99 MPa	6,57 MPa	41,91 %
	50 m	5,33 MPa	4,34 MPa	16,59 %
	66 m	0,00 MPa	0,00 MPa	-

Cabo I, σ_{cp}	0 m	-7,12 MPa	-3,18 MPa	55,34 %
	17 m	-20,07 MPa	-10,14 MPa	49,48 %
	33 m	-16,42 MPa	-10,00 MPa	56,21 %
	50 m	-21,72 MPa	-10,05 MPa	40,15 %
	66 m	-7,12 MPa	-3,18 MPa	55,34 %
Cabo II, σ_{cg}	0 m	0,00 MPa	0,00 MPa	-
	17 m	4,83 MPa	6,01 MPa	19,63 %
	33 m	4,99 MPa	6,57 MPa	41,91 %
	50 m	5,33 MPa	6,39 MPa	16,59 %
	66 m	0,00 MPa	0,00 MPa	-
Cabo II, σ_{cp}	0 m	-7,12 MPa	-3,18 MPa	55,34 %
	17 m	-20,07 MPa	-10,14 MPa	49,48 %
	33 m	-16,42 MPa	-10,00 MPa	40,15 %
	50 m	-21,72 MPa	-13,00 MPa	55,34 %
	66 m	-7,12 MPa	-3,18 MPa	-

Embora as diferenças sejam justificáveis em determinados contextos, elas indicam áreas de melhoria no TendonTech, nomeadamente na inclusão de parâmetros que influenciam a precisão. A consideração da reentrada e do comprimento de cabo dentro do macaco contribuiria para aumentar a fiabilidade do programa. A validação contínua com ferramentas consolidadas, como o ESTICA 2020 [6], é fundamental para garantir a robustez do TendonTech.

Capítulo 5

5. Conclusões e trabalhos futuros

5.1. Conclusões

A presente dissertação teve como objetivo o desenvolvimento do programa TendonTech, uma ferramenta de cálculo automático para a modelação de cabos em vigas de betão pré-esforçado, desenvolvida em Python. Esta linguagem, conhecida pela sua versatilidade e facilidade de aprendizagem, oferece uma ampla gama de bibliotecas essenciais para manipulação e visualização de dados, destacando-se o OpenSeesPy no contexto do presente projeto.

O TendonTech realiza automaticamente diversas etapas do processo de modelação e análise estrutural, como a geração das equações dos troços do traçado do cabo entre pontos de referência, com base nas coordenadas longitudinais inseridas no ficheiro de dados. Para além disso, implementa a sequência de tesagem dos cabos, permitindo especificar a ordem de tesagem e a força aplicada em cada etapa. No que diz respeito ao cálculo estrutural, o programa utiliza o OpenSees para determinar os esforços nos cabos e na viga, considerando diferentes combinações de carregamento e identificando automaticamente as secções críticas.

A abordagem adotada demonstra a viabilidade da elaboração de software de análise estrutural por parte de estudantes de engenharia civil, independentemente da sua formação específica em informática. A programação, quando estruturada de forma clara e metódica, permite o desenvolvimento de ferramentas computacionais robustas, capazes de apoiar a prática profissional e académica nos vários domínios da engenharia.

Os resultados obtidos demonstram o potencial do TendonTech, tendo sido identificadas algumas discrepâncias em determinados valores. Estas diferenças evidenciam a necessidade de ajustes no programa, incluindo a consideração de parâmetros adicionais, como a reentrada do cabo no macaco, a definição de diferentes traçados para os cabos e a análise de sistemas estruturais alternativos. A validação contínua do programa, recorrendo a ferramentas consolidadas, é essencial para garantir que o TendonTech cumpra eficazmente os requisitos estruturais e se alinhe com as normas e práticas estabelecidas no dimensionamento de vigas pré-esforçadas.

O TendonTech revela-se um contributo relevante para o ensino do Betão Pré-esforçado, proporcionando aos estudantes uma ferramenta adicional para a compreensão dos princípios fundamentais da modelação de cabos e análise estrutural. No contexto profissional, a aplicação do programa demonstra potencial para otimizar a modelação de cabos e auxiliar no dimensionamento de estruturas pré-esforçadas, contribuindo para soluções mais eficientes e fundamentadas no cálculo estrutural.

5.2. Recomendações para trabalhos futuros

Para aprimorar a fiabilidade e precisão do TendonTech, recomenda-se a incorporação de parâmetros adicionais de modo a quantificar com maior rigor as perdas de pré-esforço. Além disso, uma validação experimental mais abrangente permitiria comparar os resultados numéricos com o comportamento real das vigas de betão pré-esforçado.

A melhoria da integração com softwares CAD é igualmente relevante, facilitando a visualização espacial dos traçados e da geometria das vigas. No que respeita à modelação dos cabos, importa otimizar a consideração da excentricidade, especialmente em vigas contínuas, onde o TendonTech ainda se baseia na excentricidade máxima.

Por fim, a ampliação da base de dados e a realização de testes com diferentes tipologias de vigas contribuirão para validar a aplicabilidade do programa a projetos reais, reforçando o seu potencial enquanto ferramenta de referência na prática profissional.

Referências

- [1] R. Ian Gilbert, N. Colin Mickleborough, e G. Ranzi, *Design of Prestressed Concrete to Eurocode 2 Second Edition*. 2017.
- [2] E. G. Nawy, *PRESTRESSED CONCRETE A Fundamental Approach*. 2009.
- [3] A. E. Naaman, *PRESTRESSED CONCRETE ANALYSIS AND DESIGN Fundamentals*. 2004.
- [4] M. R. Ehsani e J. R. Blewitt, «Design Curves for Tendon Profile in Prestressed Concrete Beams», 1986.
- [5] «Eurocódigo 2 – Projecto de estruturas de betão Parte 1-1: Regras gerais e regras para edifícios», 2010.
- [6] J. Fonseca, «ESTICA 2020 - Programa de cálculo automático de estruturas de betão pré-esforçado da biblioteca de software de PENCIL Engenharia Lda», 2020, *Porto*.
- [7] A. K. Sengupta e P. Devdas Menon, «Prestressed Concrete Structures».
- [8] M. Pajari, «Parabolic tendons in prestressed concrete – How accurate are equivalent loads?», *Rakenteiden Mekaniikka*, vol. 54, n. 3, pp. 117–129, 2021, doi: 10.23998/rm.100697.
- [9] «fib Model Code for Concrete Structures 2010», 2010.
- [10] «ISO 1920-3 Testing of concrete-Part 3: Making and curing test specimens», 2019. [Em linha]. Disponível em: www.iso.org
- [11] Dr. C. Caprani, «Civil Engineering Design (1) Prestressed Concrete», 2006. Acedido: 2 de Setembro de 2024. [Em linha]. Disponível em: <https://www.colincaprani.com/files/notes/CED1/PSC%20Notes.pdf>
- [12] «EN 10138-1:2000. Prestressing steels - Part 1: General requirements», Set. 2000.
- [13] D. Holding GmbH, «European Technical Assessment. Post-Tensioning Systems», Jun. 2018. [Em linha]. Disponível em: www.oib.or.at
- [14] C. Marchão e J. Appleton, «BETÃO ARMADO E PRÉ-ESFORÇADO II MÓDULO 1-PRÉ-ESFORÇO», 2004.
- [15] W. Podolny Jr, «UNDERSTANDING THE LOSSES IN PRESTRESSING», Out. 1969.
- [16] F. , F. G. L. S. M. H. McKenna, «Open System for Earthquake Engineering Simulation», 2000, *University of California, Berkeley*.
- [17] «Python Release Python 3.8.0 | Python.org». Acedido: 9 de Fevereiro de 2025. [Em linha]. Disponível em: https://www.python.org/downloads/release/python-380/?locale=pt_PT

- [18] «Download Now | Anaconda». Acedido: 9 de Fevereiro de 2025. [Em linha]. Disponível em: <https://www.anaconda.com/download/success>
- [19] G. Van Rossum e Python development team, «Python Tutorial Release 3.13.1», Jan. 2025. Acedido: 2 de Janeiro de 2025. [Em linha]. Disponível em: <https://docs.python.org/3/download.html>
- [20] «OpenSeesWiki». Acedido: 7 de Janeiro de 2025. [Em linha]. Disponível em: https://opensees.ist.berkeley.edu/wiki/index.php?title=Main_Page
- [21] «Open System for Earthquake Engineering Simulation - Home Page». Acedido: 7 de Janeiro de 2025. [Em linha]. Disponível em: <https://opensees.berkeley.edu/>
- [22] «openseesPy - DesignSafe User Guide». Acedido: 7 de Janeiro de 2025. [Em linha]. Disponível em: <https://designsafe-ci.org/user-guide/tools/simulation/opensees/openseesPy/>

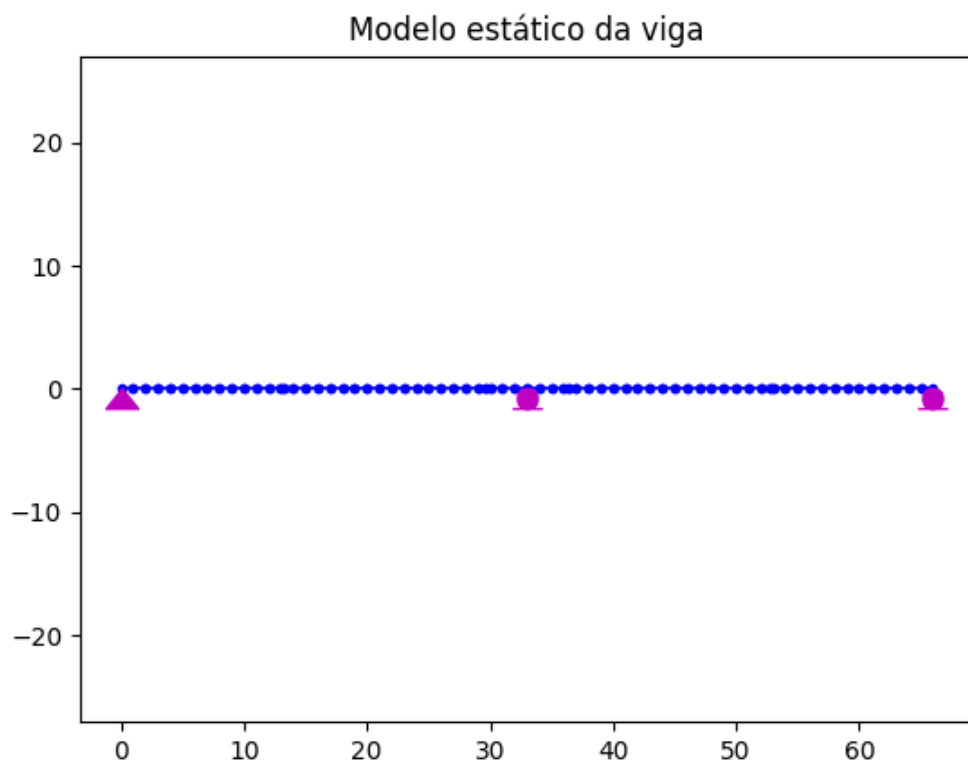
Anexo 1

A. Relatório de cálculo: aplicação prática do programa

***** TendonTech 1.0 2024 *****

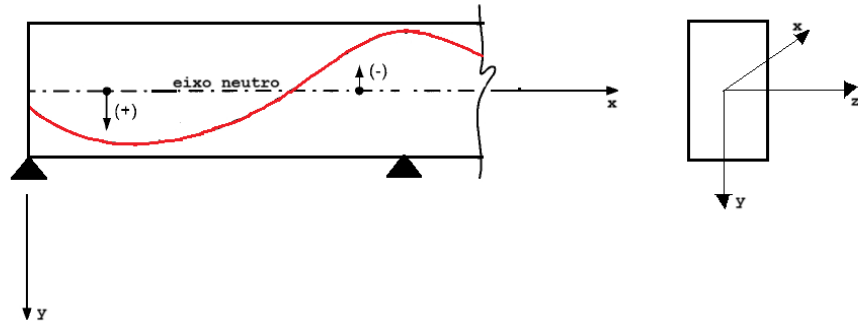
Projeto de Aplicação de Pré-Esforço

Viga pré-esforçada: 33m no vão 1 e 33m no vão 2, com um comprimento total de 66.0m.



RELATÓRIO DE CÁLCULO

Convenção de sinais



Caraterísticas do Betão

Classe: C30/37

Módulo de elasticidade, E_{cm} : 33 GPa

Coeficiente de fluência, $\phi(t, t_0)$: 2.16

Extensão total de retração, ϵ_{cs} : 0.2 ‰

Propriedades Geométricas da Secção Transversal de Betão

Área, A_c : 1.24 m²

Momento de inércia, I_z : 0.4 m⁴

Altura, h : 1.8 m

Perímetro, u : 8.8 m

Distância do eixo neutro à fibra inferior, v_i : 1.177 m

Distância do eixo neutro à fibra superior, v_s : -0.623 m

Módulo de Resistência da fibra inferior, z_i : 0.34 m³

Módulo de Resistência da fibra superior, z_s : -0.642 m³

Distância do centro geométrico ao limite superior do Núcleo Central, k_i : 0.274 m

Distância do centro geométrico ao limite inferior do Núcleo Central, k_s : -0.518 m

Excentricidade limite (vãos), e_{lim} : 1.027 m

Excentricidade limite (apoio), e_{lim} : -0.473 m

Caraterísticas do Sistema de Pré-Esforço

Coefficiente de Atrito, μ : 0.2

Desvio Angular Parasita, k : 0.01 rad/m

Reentrada das Cunhas nas ancoragem, Δs_1 : 6.0 mm

Caraterísticas do Aço de Pré-Esforço

Tensão de rotura, f_{puk} : 1860 MPa

Módulo de elasticidade, E_p : 195 MPa

Relaxação (1000h após aplicação de pré-esforço), ρ_{1000} : 2.5 %

Área nominal: 1.5 cm²

Caraterísticas dos cabos de Pré-esforço e Força de Tensionamento

Ordem	Tipo	Cabo	Extr.	Área [cm ²]	L [m]	Bainha [mm]	P _{máx} [kN]	ΔL [mm]
1°	1	1	1	30.0	66.257	100	4464	426
2°	1	1	2	30.0	33.129	100	4464	45
3°	1	2	1	30.0	66.257	100	4464	426
4°	1	2	2	30.0	33.129	100	4464	45

Nota: O Tipo 0 é referente a uma sequência de traçados retos; o Tipo 1 é referente a uma sequência de traçados parabólicos.

Cargas e Fatores de Redução

Carga permanente, g : -31 kN/m

Carga variável, q : 0 kN/m

ψ_2 : 0.2

Perdas iniciais de Pré-Esforço

CABO I: Pré-Esforço após perdas por atrito, P_μ

Secção	x [m]	Alfa [rad]	dP _{mu} [kN]	P _{mu} [kN]	d_mu [%]
S1	0.0	0.0	0	4464	0.0
S2	1.0	0.012	20	4444	0.45
S3	2.0	0.0239	39	4425	0.87

S4	3.0	0.0359	58	4406	1.3
S5	4.0	0.0479	78	4386	1.75
S6	5.0	0.0598	97	4367	2.17
S7	6.0	0.0717	116	4348	2.6
S8	7.0	0.0836	135	4329	3.02
S9	8.0	0.0955	154	4310	3.45
S10	9.0	0.1074	173	4291	3.88
S11	10.0	0.1192	191	4273	4.28
S12	11.0	0.131	210	4254	4.7
S13	12.0	0.1427	228	4236	5.11
S14	13.0	0.1544	247	4217	5.53
S15	13.2	0.1544	249	4215	5.58
S16	14.0	0.1639	263	4201	5.89
S17	15.0	0.1733	280	4184	6.27
S18	16.0	0.1828	296	4168	6.63
S19	17.0	0.1923	312	4152	6.99
S20	18.0	0.2017	328	4136	7.35
S21	19.0	0.2112	344	4120	7.71
S22	20.0	0.2206	360	4104	8.06
S23	21.0	0.23	376	4088	8.42
S24	22.0	0.2394	392	4072	8.78
S25	23.0	0.2488	408	4056	9.14
S26	24.0	0.2582	423	4041	9.48
S27	25.0	0.2676	439	4025	9.83
S28	26.0	0.2769	455	4009	10.19
S29	27.0	0.2862	470	3994	10.53
S30	28.0	0.2955	485	3979	10.86
S31	29.0	0.3048	501	3963	11.22
S32	29.7	0.3048	506	3958	11.34
S33	30.0	0.3553	548	3916	12.28

S34	31.0	0.4055	595	3869	13.33
S35	32.0	0.4552	641	3823	14.36
S36	33.0	0.4552	649	3815	14.54
S37	34.0	0.5057	641	3823	14.36
S38	35.0	0.5559	595	3869	13.33
S39	36.0	0.6056	548	3916	12.28
S40	36.3	0.6056	506	3958	11.34
S41	37.0	0.6151	501	3963	11.22
S42	38.0	0.6245	485	3979	10.86
S43	39.0	0.634	470	3994	10.53
S44	40.0	0.6435	455	4009	10.19
S45	41.0	0.6529	439	4025	9.83
S46	42.0	0.6624	423	4041	9.48
S47	43.0	0.6718	408	4056	9.14
S48	44.0	0.6812	392	4072	8.78
S49	45.0	0.6906	376	4088	8.42
S50	46.0	0.7	360	4104	8.06
S51	47.0	0.7094	344	4120	7.71
S52	48.0	0.7188	328	4136	7.35
S53	49.0	0.7281	312	4152	6.99
S54	50.0	0.7374	296	4168	6.63
S55	51.0	0.7467	280	4184	6.27
S56	52.0	0.756	263	4201	5.89
S57	52.8	0.756	249	4215	5.58
S58	53.0	0.768	247	4217	5.53
S59	54.0	0.7799	228	4236	5.11
S60	55.0	0.7919	210	4254	4.7
S61	56.0	0.8039	191	4273	4.28
S62	57.0	0.8158	173	4291	3.88
S63	58.0	0.8277	154	4310	3.45

S64	59.0	0.8396	135	4329	3.02
S65	60.0	0.8515	116	4348	2.6
S66	61.0	0.8634	97	4367	2.17
S67	62.0	0.8752	78	4386	1.75
S68	63.0	0.887	58	4406	1.3
S69	64.0	0.8987	39	4425	0.87
S70	65.0	0.9104	20	4444	0.45
S71	66.0	0.9104	0	4464	0.0

CABO I: Pré-Esforço após perdas por reentrada das cunhas, Psl

Secção	x [m]	dPsl [kN]	Psl [kN]	d_sl [%]
S1	0.0	520	3944	11.65
S2	1.0	482	3962	11.25
S3	2.0	443	3982	10.79
S4	3.0	404	4002	10.35
S5	4.0	366	4020	9.95
S6	5.0	327	4040	9.5
S7	6.0	289	4059	9.07
S8	7.0	250	4079	8.62
S9	8.0	212	4098	8.2
S10	9.0	173	4118	7.76
S11	10.0	135	4138	7.3
S12	11.0	96	4158	6.85
S13	12.0	58	4178	6.41
S14	13.0	19	4198	5.96
S15	13.2	11	4204	5.83
S16	14.0	0	4201	5.89
S17	15.0	0	4184	6.27
S18	16.0	0	4168	6.63
S19	17.0	0	4152	6.99
S20	18.0	0	4136	7.35

S21	19.0	0	4120	7.71
S22	20.0	0	4104	8.06
S23	21.0	0	4088	8.42
S24	22.0	0	4072	8.78
S25	23.0	0	4056	9.14
S26	24.0	0	4041	9.48
S27	25.0	0	4025	9.83
S28	26.0	0	4009	10.19
S29	27.0	0	3994	10.53
S30	28.0	0	3979	10.86
S31	29.0	0	3963	11.22
S32	29.7	0	3958	11.34
S33	30.0	0	3916	12.28
S34	31.0	0	3869	13.33
S35	32.0	0	3823	14.36
S36	33.0	0	3815	14.54
S37	34.0	0	3823	14.36
S38	35.0	0	3869	13.33
S39	36.0	0	3916	12.28
S40	36.3	0	3958	11.34
S41	37.0	0	3963	11.22
S42	38.0	0	3979	10.86
S43	39.0	0	3994	10.53
S44	40.0	0	4009	10.19
S45	41.0	0	4025	9.83
S46	42.0	0	4041	9.48
S47	43.0	0	4056	9.14
S48	44.0	0	4072	8.78
S49	45.0	0	4088	8.42
S50	46.0	0	4104	8.06

S51	47.0	0	4120	7.71
S52	48.0	0	4136	7.35
S53	49.0	0	4152	6.99
S54	50.0	0	4168	6.63
S55	51.0	0	4184	6.27
S56	52.0	0	4201	5.89
S57	52.8	11	4204	5.83
S58	53.0	19	4198	5.96
S59	54.0	58	4178	6.41
S60	55.0	96	4158	6.85
S61	56.0	135	4138	7.3
S62	57.0	173	4118	7.76
S63	58.0	212	4098	8.2
S64	59.0	250	4079	8.62
S65	60.0	289	4059	9.07
S66	61.0	327	4040	9.5
S67	62.0	366	4020	9.95
S68	63.0	404	4002	10.35
S69	64.0	443	3982	10.79
S70	65.0	482	3962	11.25
S71	66.0	520	3944	11.65

CABO I: Pré-Esforço após perdas por deformação elástica do betão, Pel

Secção	x [m]	Ps1 [kN]	S_cg [MPa]	S_cp [MPa]	eps_c [mm/m]	dPel [kN]	Pel [kN]	d_el [%]
S1	0.0	3944	-0.0	-3.18	-0.1	58	3886	12.95
S2	1.0	3962	1.08	-4.71	-0.11	64	3898	12.68
S3	2.0	3982	2.08	-6.1	-0.12	70	3912	12.36
S4	3.0	4002	2.98	-7.35	-0.13	76	3926	12.05
S5	4.0	4020	3.79	-8.46	-0.14	82	3938	11.79
S6	5.0	4040	4.51	-9.43	-0.15	88	3952	11.47

S7	6.0	4059	5.13	-10.25	-0.16	94	3965	11.18
S8	7.0	4079	5.67	-10.94	-0.16	94	3985	10.73
S9	8.0	4098	6.12	-11.48	-0.16	94	4004	10.31
S10	9.0	4118	6.47	-11.89	-0.17	99	4019	9.98
S11	10.0	4138	6.73	-12.15	-0.17	99	4039	9.52
S12	11.0	4158	6.9	-12.26	-0.16	94	4064	8.96
S13	12.0	4178	6.98	-12.24	-0.16	94	4084	8.52
S14	13.0	4198	6.97	-12.07	-0.16	94	4104	8.07
S15	13.2	4204	6.96	-12.04	-0.15	88	4116	7.8
S16	14.0	4201	6.87	-11.79	-0.15	88	4113	7.86
S17	15.0	4184	6.67	-11.36	-0.14	82	4102	8.11
S18	16.0	4168	6.39	-10.81	-0.13	76	4092	8.33
S19	17.0	4152	6.01	-10.14	-0.13	76	4076	8.69
S20	18.0	4136	5.54	-9.36	-0.12	70	4066	8.92
S21	19.0	4120	4.99	-8.46	-0.11	64	4056	9.14
S22	20.0	4104	4.34	-7.45	-0.09	53	4051	9.25
S23	21.0	4088	3.59	-6.32	-0.08	47	4041	9.47
S24	22.0	4072	2.76	-5.08	-0.07	41	4031	9.7
S25	23.0	4056	1.84	-3.73	-0.06	35	4021	9.92
S26	24.0	4041	0.82	-2.26	-0.04	23	4018	10.0
S27	25.0	4025	-0.29	-0.69	-0.03	18	4007	10.23
S28	26.0	4009	-1.48	1.0	-0.01	6	4003	10.32
S29	27.0	3994	-2.77	2.8	0.0	0	3994	10.53
S30	28.0	3979	-4.15	4.71	0.02	12	3967	11.13
S31	29.0	3963	-5.62	6.73	0.03	18	3945	11.62
S32	29.7	3958	3.55	-9.18	-0.17	99	3859	13.56
S33	30.0	3916	3.8	-9.37	-0.17	99	3817	14.5
S34	31.0	3869	4.68	-9.77	-0.16	94	3775	15.44
S35	32.0	3823	5.6	-9.87	-0.13	76	3747	16.06
S36	33.0	3815	6.57	-10.0	-0.1	58	3757	15.84

S37	34.0	3823	5.6	-9.48	-0.12	70	3753	15.93
S38	35.0	3869	4.68	-8.69	-0.12	70	3799	14.9
S39	36.0	3916	3.8	-7.61	-0.12	70	3846	13.85
S40	36.3	3958	3.55	-7.31	-0.11	64	3894	12.77
S41	37.0	3963	-5.62	3.18	-0.07	41	3922	12.14
S42	38.0	3979	-4.15	1.26	-0.09	53	3926	12.05
S43	39.0	3994	-2.77	-0.54	-0.1	58	3936	11.83
S44	40.0	4009	-1.48	-2.23	-0.11	64	3945	11.62
S45	41.0	4025	-0.29	-3.81	-0.12	70	3955	11.4
S46	42.0	4041	0.82	-5.28	-0.14	82	3959	11.32
S47	43.0	4056	1.84	-6.64	-0.15	88	3968	11.11
S48	44.0	4072	2.76	-7.89	-0.16	94	3978	10.89
S49	45.0	4088	3.59	-9.02	-0.17	99	3989	10.64
S50	46.0	4104	4.34	-10.05	-0.17	99	4005	10.28
S51	47.0	4120	4.99	-10.96	-0.18	105	4015	10.06
S52	48.0	4136	5.54	-11.75	-0.19	111	4025	9.84
S53	49.0	4152	6.01	-12.43	-0.2	117	4035	9.61
S54	50.0	4168	6.39	-13.0	-0.2	117	4051	9.25
S55	51.0	4184	6.67	-13.46	-0.21	123	4061	9.03
S56	52.0	4201	6.87	-13.8	-0.21	123	4078	8.65
S57	52.8	4204	6.96	-14.07	-0.22	129	4075	8.72
S58	53.0	4198	6.97	-14.1	-0.22	129	4069	8.85
S59	54.0	4178	6.98	-14.11	-0.22	129	4049	9.3
S60	55.0	4158	6.9	-13.98	-0.22	129	4029	9.74
S61	56.0	4138	6.73	-13.7	-0.21	123	4015	10.06
S62	57.0	4118	6.47	-13.28	-0.21	123	3995	10.52
S63	58.0	4098	6.12	-12.71	-0.2	117	3981	10.82
S64	59.0	4079	5.67	-12.0	-0.19	111	3968	11.11
S65	60.0	4059	5.13	-11.14	-0.18	105	3954	11.42
S66	61.0	4040	4.51	-10.14	-0.17	99	3941	11.72

S67	62.0	4020	3.79	-9.0	-0.16	94	3926	12.06
S68	63.0	4002	2.98	-7.72	-0.14	82	3920	12.19
S69	64.0	3982	2.08	-6.3	-0.13	76	3906	12.49
S70	65.0	3962	1.08	-4.74	-0.11	64	3898	12.68
S71	66.0	3944	0.0	-3.18	-0.1	58	3886	12.95

Obs: Perdas por deformação elástica do betão no cabo 1 devidas ao esticamento do cabo 2.

CABO II: Pré-Esforço após perdas por atrito, P_{μ}

Secção	x [m]	Alfa [rad]	dPmu [kN]	Pmu [kN]	d_mu [%]
S1	0.0	0.0	0	4464	0.0
S2	1.0	0.012	20	4444	0.45
S3	2.0	0.0239	39	4425	0.87
S4	3.0	0.0359	58	4406	1.3
S5	4.0	0.0479	78	4386	1.75
S6	5.0	0.0598	97	4367	2.17
S7	6.0	0.0717	116	4348	2.6
S8	7.0	0.0836	135	4329	3.02
S9	8.0	0.0955	154	4310	3.45
S10	9.0	0.1074	173	4291	3.88
S11	10.0	0.1192	191	4273	4.28
S12	11.0	0.131	210	4254	4.7
S13	12.0	0.1427	228	4236	5.11
S14	13.0	0.1544	247	4217	5.53
S15	13.2	0.1544	249	4215	5.58
S16	14.0	0.1639	263	4201	5.89
S17	15.0	0.1733	280	4184	6.27
S18	16.0	0.1828	296	4168	6.63
S19	17.0	0.1923	312	4152	6.99
S20	18.0	0.2017	328	4136	7.35
S21	19.0	0.2112	344	4120	7.71

S22	20.0	0.2206	360	4104	8.06
S23	21.0	0.23	376	4088	8.42
S24	22.0	0.2394	392	4072	8.78
S25	23.0	0.2488	408	4056	9.14
S26	24.0	0.2582	423	4041	9.48
S27	25.0	0.2676	439	4025	9.83
S28	26.0	0.2769	455	4009	10.19
S29	27.0	0.2862	470	3994	10.53
S30	28.0	0.2955	485	3979	10.86
S31	29.0	0.3048	501	3963	11.22
S32	29.7	0.3048	506	3958	11.34
S33	30.0	0.3553	548	3916	12.28
S34	31.0	0.4055	595	3869	13.33
S35	32.0	0.4552	641	3823	14.36
S36	33.0	0.4552	649	3815	14.54
S37	34.0	0.5057	641	3823	14.36
S38	35.0	0.5559	595	3869	13.33
S39	36.0	0.6056	548	3916	12.28
S40	36.3	0.6056	506	3958	11.34
S41	37.0	0.6151	501	3963	11.22
S42	38.0	0.6245	485	3979	10.86
S43	39.0	0.634	470	3994	10.53
S44	40.0	0.6435	455	4009	10.19
S45	41.0	0.6529	439	4025	9.83
S46	42.0	0.6624	423	4041	9.48
S47	43.0	0.6718	408	4056	9.14
S48	44.0	0.6812	392	4072	8.78
S49	45.0	0.6906	376	4088	8.42
S50	46.0	0.7	360	4104	8.06
S51	47.0	0.7094	344	4120	7.71

S52	48.0	0.7188	328	4136	7.35
S53	49.0	0.7281	312	4152	6.99
S54	50.0	0.7374	296	4168	6.63
S55	51.0	0.7467	280	4184	6.27
S56	52.0	0.756	263	4201	5.89
S57	52.8	0.756	249	4215	5.58
S58	53.0	0.768	247	4217	5.53
S59	54.0	0.7799	228	4236	5.11
S60	55.0	0.7919	210	4254	4.7
S61	56.0	0.8039	191	4273	4.28
S62	57.0	0.8158	173	4291	3.88
S63	58.0	0.8277	154	4310	3.45
S64	59.0	0.8396	135	4329	3.02
S65	60.0	0.8515	116	4348	2.6
S66	61.0	0.8634	97	4367	2.17
S67	62.0	0.8752	78	4386	1.75
S68	63.0	0.887	58	4406	1.3
S69	64.0	0.8987	39	4425	0.87
S70	65.0	0.9104	20	4444	0.45
S71	66.0	0.9104	0	4464	0.0

CABO II: Pré-Esforço após perdas por reentrada das cunhas, Psl

Secção	x [m]	dPsl [kN]	Psl [kN]	d_sl [%]
S1	0.0	520	3944	11.65
S2	1.0	482	3962	11.25
S3	2.0	443	3982	10.79
S4	3.0	404	4002	10.35
S5	4.0	366	4020	9.95
S6	5.0	327	4040	9.5
S7	6.0	289	4059	9.07
S8	7.0	250	4079	8.62

S9	8.0	212	4098	8.2
S10	9.0	173	4118	7.76
S11	10.0	135	4138	7.3
S12	11.0	96	4158	6.85
S13	12.0	58	4178	6.41
S14	13.0	19	4198	5.96
S15	13.2	11	4204	5.83
S16	14.0	0	4201	5.89
S17	15.0	0	4184	6.27
S18	16.0	0	4168	6.63
S19	17.0	0	4152	6.99
S20	18.0	0	4136	7.35
S21	19.0	0	4120	7.71
S22	20.0	0	4104	8.06
S23	21.0	0	4088	8.42
S24	22.0	0	4072	8.78
S25	23.0	0	4056	9.14
S26	24.0	0	4041	9.48
S27	25.0	0	4025	9.83
S28	26.0	0	4009	10.19
S29	27.0	0	3994	10.53
S30	28.0	0	3979	10.86
S31	29.0	0	3963	11.22
S32	29.7	0	3958	11.34
S33	30.0	0	3916	12.28
S34	31.0	0	3869	13.33
S35	32.0	0	3823	14.36
S36	33.0	0	3815	14.54
S37	34.0	0	3823	14.36
S38	35.0	0	3869	13.33

S39	36.0	0	3916	12.28
S40	36.3	0	3958	11.34
S41	37.0	0	3963	11.22
S42	38.0	0	3979	10.86
S43	39.0	0	3994	10.53
S44	40.0	0	4009	10.19
S45	41.0	0	4025	9.83
S46	42.0	0	4041	9.48
S47	43.0	0	4056	9.14
S48	44.0	0	4072	8.78
S49	45.0	0	4088	8.42
S50	46.0	0	4104	8.06
S51	47.0	0	4120	7.71
S52	48.0	0	4136	7.35
S53	49.0	0	4152	6.99
S54	50.0	0	4168	6.63
S55	51.0	0	4184	6.27
S56	52.0	0	4201	5.89
S57	52.8	11	4204	5.83
S58	53.0	19	4198	5.96
S59	54.0	58	4178	6.41
S60	55.0	96	4158	6.85
S61	56.0	135	4138	7.3
S62	57.0	173	4118	7.76
S63	58.0	212	4098	8.2
S64	59.0	250	4079	8.62
S65	60.0	289	4059	9.07
S66	61.0	327	4040	9.5
S67	62.0	366	4020	9.95
S68	63.0	404	4002	10.35

S69	64.0	443	3982	10.79
S70	65.0	482	3962	11.25
S71	66.0	520	3944	11.65

Pré-Esforço após Perdas Instantâneas, Pm0

Cabo I:

Secção	x [m]	Pm0 [kN]	S_cg [MPa]	S_cp [MPa]	eps_c [mm/m]	d_el [%]
S1	0.0	3886	-0.0	-3.18	-0.1	12.95
S2	1.0	3898	1.08	-4.71	-0.11	12.68
S3	2.0	3912	2.08	-6.1	-0.12	12.36
S4	3.0	3926	2.98	-7.35	-0.13	12.05
S5	4.0	3938	3.79	-8.46	-0.14	11.79
S6	5.0	3952	4.51	-9.43	-0.15	11.47
S7	6.0	3965	5.13	-10.25	-0.16	11.18
S8	7.0	3985	5.67	-10.94	-0.16	10.73
S9	8.0	4004	6.12	-11.48	-0.16	10.31
S10	9.0	4019	6.47	-11.89	-0.17	9.98
S11	10.0	4039	6.73	-12.15	-0.17	9.52
S12	11.0	4064	6.9	-12.26	-0.16	8.96
S13	12.0	4084	6.98	-12.24	-0.16	8.52
S14	13.0	4104	6.97	-12.07	-0.16	8.07
S15	13.2	4116	6.96	-12.04	-0.15	7.8
S16	14.0	4113	6.87	-11.79	-0.15	7.86
S17	15.0	4102	6.67	-11.36	-0.14	8.11
S18	16.0	4092	6.39	-10.81	-0.13	8.33
S19	17.0	4076	6.01	-10.14	-0.13	8.69
S20	18.0	4066	5.54	-9.36	-0.12	8.92
S21	19.0	4056	4.99	-8.46	-0.11	9.14
S22	20.0	4051	4.34	-7.45	-0.09	9.25
S23	21.0	4041	3.59	-6.32	-0.08	9.47

S24	22.0	4031	2.76	-5.08	-0.07	9.7
S25	23.0	4021	1.84	-3.73	-0.06	9.92
S26	24.0	4018	0.82	-2.26	-0.04	10.0
S27	25.0	4007	-0.29	-0.69	-0.03	10.23
S28	26.0	4003	-1.48	1.0	-0.01	10.32
S29	27.0	3994	-2.77	2.8	0.0	10.53
S30	28.0	3967	-4.15	4.71	0.02	11.13
S31	29.0	3945	-5.62	6.73	0.03	11.62
S32	29.7	3859	3.55	-9.18	-0.17	13.56
S33	30.0	3817	3.8	-9.37	-0.17	14.5
S34	31.0	3775	4.68	-9.77	-0.16	15.44
S35	32.0	3747	5.6	-9.87	-0.13	16.06
S36	33.0	3757	6.57	-10.0	-0.1	15.84
S37	34.0	3753	5.6	-9.48	-0.12	15.93
S38	35.0	3799	4.68	-8.69	-0.12	14.9
S39	36.0	3846	3.8	-7.61	-0.12	13.85
S40	36.3	3894	3.55	-7.31	-0.11	12.77
S41	37.0	3922	-5.62	3.18	-0.07	12.14
S42	38.0	3926	-4.15	1.26	-0.09	12.05
S43	39.0	3936	-2.77	-0.54	-0.1	11.83
S44	40.0	3945	-1.48	-2.23	-0.11	11.62
S45	41.0	3955	-0.29	-3.81	-0.12	11.4
S46	42.0	3959	0.82	-5.28	-0.14	11.32
S47	43.0	3968	1.84	-6.64	-0.15	11.11
S48	44.0	3978	2.76	-7.89	-0.16	10.89
S49	45.0	3989	3.59	-9.02	-0.17	10.64
S50	46.0	4005	4.34	-10.05	-0.17	10.28
S51	47.0	4015	4.99	-10.96	-0.18	10.06
S52	48.0	4025	5.54	-11.75	-0.19	9.84
S53	49.0	4035	6.01	-12.43	-0.2	9.61

S54	50.0	4051	6.39	-13.0	-0.2	9.25
S55	51.0	4061	6.67	-13.46	-0.21	9.03
S56	52.0	4078	6.87	-13.8	-0.21	8.65
S57	52.8	4075	6.96	-14.07	-0.22	8.72
S58	53.0	4069	6.97	-14.1	-0.22	8.85
S59	54.0	4049	6.98	-14.11	-0.22	9.3
S60	55.0	4029	6.9	-13.98	-0.22	9.74
S61	56.0	4015	6.73	-13.7	-0.21	10.06
S62	57.0	3995	6.47	-13.28	-0.21	10.52
S63	58.0	3981	6.12	-12.71	-0.2	10.82
S64	59.0	3968	5.67	-12.0	-0.19	11.11
S65	60.0	3954	5.13	-11.14	-0.18	11.42
S66	61.0	3941	4.51	-10.14	-0.17	11.72
S67	62.0	3926	3.79	-9.0	-0.16	12.06
S68	63.0	3920	2.98	-7.72	-0.14	12.19
S69	64.0	3906	2.08	-6.3	-0.13	12.49
S70	65.0	3898	1.08	-4.74	-0.11	12.68
S71	66.0	3886	0.0	-3.18	-0.1	12.95

Cabo II:

Secção	x [m]	Pm0 [kN]	S_cg [MPa]	S_cp [MPa]	eps_c [mm/m]	d_sl [%]
S1	0.0	3944	-0.0	-3.18	-0.1	11.65
S2	1.0	3962	1.08	-4.71	-0.11	11.25
S3	2.0	3982	2.08	-6.1	-0.12	10.79
S4	3.0	4002	2.98	-7.35	-0.13	10.35
S5	4.0	4020	3.79	-8.46	-0.14	9.95
S6	5.0	4040	4.51	-9.43	-0.15	9.5
S7	6.0	4059	5.13	-10.25	-0.16	9.07
S8	7.0	4079	5.67	-10.94	-0.16	8.62
S9	8.0	4098	6.12	-11.48	-0.16	8.2

S10	9.0	4118	6.47	-11.89	-0.17	7.76
S11	10.0	4138	6.73	-12.15	-0.17	7.3
S12	11.0	4158	6.9	-12.26	-0.16	6.85
S13	12.0	4178	6.98	-12.24	-0.16	6.41
S14	13.0	4198	6.97	-12.07	-0.16	5.96
S15	13.2	4204	6.96	-12.04	-0.15	5.83
S16	14.0	4201	6.87	-11.79	-0.15	5.89
S17	15.0	4184	6.67	-11.36	-0.14	6.27
S18	16.0	4168	6.39	-10.81	-0.13	6.63
S19	17.0	4152	6.01	-10.14	-0.13	6.99
S20	18.0	4136	5.54	-9.36	-0.12	7.35
S21	19.0	4120	4.99	-8.46	-0.11	7.71
S22	20.0	4104	4.34	-7.45	-0.09	8.06
S23	21.0	4088	3.59	-6.32	-0.08	8.42
S24	22.0	4072	2.76	-5.08	-0.07	8.78
S25	23.0	4056	1.84	-3.73	-0.06	9.14
S26	24.0	4041	0.82	-2.26	-0.04	9.48
S27	25.0	4025	-0.29	-0.69	-0.03	9.83
S28	26.0	4009	-1.48	1.0	-0.01	10.19
S29	27.0	3994	-2.77	2.8	0.0	10.53
S30	28.0	3979	-4.15	4.71	0.02	10.86
S31	29.0	3963	-5.62	6.73	0.03	11.22
S32	29.7	3958	3.55	-9.18	-0.17	11.34
S33	30.0	3916	3.8	-9.37	-0.17	12.28
S34	31.0	3869	4.68	-9.77	-0.16	13.33
S35	32.0	3823	5.6	-9.87	-0.13	14.36
S36	33.0	3815	6.57	-10.0	-0.1	14.54
S37	34.0	3823	5.6	-9.48	-0.12	14.36
S38	35.0	3869	4.68	-8.69	-0.12	13.33
S39	36.0	3916	3.8	-7.61	-0.12	12.28

S40	36.3	3958	3.55	-7.31	-0.11	11.34
S41	37.0	3963	-5.62	3.18	-0.07	11.22
S42	38.0	3979	-4.15	1.26	-0.09	10.86
S43	39.0	3994	-2.77	-0.54	-0.1	10.53
S44	40.0	4009	-1.48	-2.23	-0.11	10.19
S45	41.0	4025	-0.29	-3.81	-0.12	9.83
S46	42.0	4041	0.82	-5.28	-0.14	9.48
S47	43.0	4056	1.84	-6.64	-0.15	9.14
S48	44.0	4072	2.76	-7.89	-0.16	8.78
S49	45.0	4088	3.59	-9.02	-0.17	8.42
S50	46.0	4104	4.34	-10.05	-0.17	8.06
S51	47.0	4120	4.99	-10.96	-0.18	7.71
S52	48.0	4136	5.54	-11.75	-0.19	7.35
S53	49.0	4152	6.01	-12.43	-0.2	6.99
S54	50.0	4168	6.39	-13.0	-0.2	6.63
S55	51.0	4184	6.67	-13.46	-0.21	6.27
S56	52.0	4201	6.87	-13.8	-0.21	5.89
S57	52.8	4204	6.96	-14.07	-0.22	5.83
S58	53.0	4198	6.97	-14.1	-0.22	5.96
S59	54.0	4178	6.98	-14.11	-0.22	6.41
S60	55.0	4158	6.9	-13.98	-0.22	6.85
S61	56.0	4138	6.73	-13.7	-0.21	7.3
S62	57.0	4118	6.47	-13.28	-0.21	7.76
S63	58.0	4098	6.12	-12.71	-0.2	8.2
S64	59.0	4079	5.67	-12.0	-0.19	8.62
S65	60.0	4059	5.13	-11.14	-0.18	9.07
S66	61.0	4040	4.51	-10.14	-0.17	9.5
S67	62.0	4020	3.79	-9.0	-0.16	9.95
S68	63.0	4002	2.98	-7.72	-0.14	10.35
S69	64.0	3982	2.08	-6.3	-0.13	10.79

S70	65.0	3962	1.08	-4.74	-0.11	11.25
S71	66.0	3944	0.0	-3.18	-0.1	11.65

Pré-Esforço após Perdas Diferidas, P_{∞}

Cabo I:

Secção	x [m]	Pm0 [kN]	dPdif [kN]	P_util [kN]	d_dif [%]
S1	0.0	3886	115	3771	15.53
S2	1.0	3898	115	3783	15.26
S3	2.0	3912	115	3797	14.94
S4	3.0	3926	115	3811	14.63
S5	4.0	3938	115	3823	14.37
S6	5.0	3952	115	3837	14.05
S7	6.0	3965	114	3851	13.73
S8	7.0	3985	114	3871	13.28
S9	8.0	4004	113	3891	12.84
S10	9.0	4019	112	3907	12.49
S11	10.0	4039	111	3928	12.01
S12	11.0	4064	109	3955	11.4
S13	12.0	4084	106	3978	10.89
S14	13.0	4104	103	4001	10.38
S15	13.2	4116	102	4014	10.08
S16	14.0	4113	102	4011	10.14
S17	15.0	4102	103	3999	10.42
S18	16.0	4092	103	3989	10.64
S19	17.0	4076	103	3973	11.0
S20	18.0	4066	104	3962	11.25
S21	19.0	4056	105	3951	11.49
S22	20.0	4051	106	3945	11.62
S23	21.0	4041	107	3934	11.87
S24	22.0	4031	108	3923	12.12

S25	23.0	4021	110	3911	12.38
S26	24.0	4018	111	3907	12.49
S27	25.0	4007	112	3895	12.74
S28	26.0	4003	113	3890	12.85
S29	27.0	3994	114	3880	13.08
S30	28.0	3967	115	3852	13.71
S31	29.0	3945	115	3830	14.2
S32	29.7	3859	115	3744	16.14
S33	30.0	3817	114	3703	17.05
S34	31.0	3775	114	3661	17.99
S35	32.0	3747	114	3633	18.61
S36	33.0	3757	114	3643	18.39
S37	34.0	3753	113	3640	18.46
S38	35.0	3799	112	3687	17.41
S39	36.0	3846	112	3734	16.36
S40	36.3	3894	112	3782	15.28
S41	37.0	3922	113	3809	14.67
S42	38.0	3926	114	3812	14.6
S43	39.0	3936	114	3822	14.38
S44	40.0	3945	114	3831	14.17
S45	41.0	3955	114	3841	13.95
S46	42.0	3959	115	3844	13.9
S47	43.0	3968	115	3853	13.69
S48	44.0	3978	115	3863	13.47
S49	45.0	3989	115	3874	13.22
S50	46.0	4005	115	3890	12.86
S51	47.0	4015	115	3900	12.64
S52	48.0	4025	115	3910	12.42
S53	49.0	4035	115	3920	12.19
S54	50.0	4051	114	3937	11.8

S55	51.0	4061	114	3947	11.58
S56	52.0	4078	112	3966	11.16
S57	52.8	4075	111	3964	11.21
S58	53.0	4069	109	3960	11.29
S59	54.0	4049	107	3942	11.7
S60	55.0	4029	104	3925	12.07
S61	56.0	4015	102	3913	12.34
S62	57.0	3995	102	3893	12.8
S63	58.0	3981	103	3878	13.13
S64	59.0	3968	106	3862	13.48
S65	60.0	3954	109	3845	13.86
S66	61.0	3941	111	3830	14.21
S67	62.0	3926	112	3814	14.57
S68	63.0	3920	113	3807	14.72
S69	64.0	3906	114	3792	15.04
S70	65.0	3898	114	3784	15.23
S71	66.0	3886	115	3771	15.53

Cabo II:

Secção	x [m]	Pm0 [kN]	dPdif [kN]	P_util [kN]	d_dif [%]
S1	0.0	3944	115	3829	14.23
S2	1.0	3962	115	3847	13.83
S3	2.0	3982	115	3867	13.37
S4	3.0	4002	115	3887	12.93
S5	4.0	4020	115	3905	12.53
S6	5.0	4040	115	3925	12.08
S7	6.0	4059	114	3945	11.62
S8	7.0	4079	114	3965	11.17
S9	8.0	4098	113	3985	10.73
S10	9.0	4118	112	4006	10.27

S11	10.0	4138	111	4027	9.79
S12	11.0	4158	109	4049	9.29
S13	12.0	4178	106	4072	8.78
S14	13.0	4198	103	4095	8.27
S15	13.2	4204	102	4102	8.11
S16	14.0	4201	102	4099	8.17
S17	15.0	4184	103	4081	8.58
S18	16.0	4168	103	4065	8.94
S19	17.0	4152	103	4049	9.3
S20	18.0	4136	104	4032	9.68
S21	19.0	4120	105	4015	10.06
S22	20.0	4104	106	3998	10.43
S23	21.0	4088	107	3981	10.82
S24	22.0	4072	108	3964	11.2
S25	23.0	4056	110	3946	11.6
S26	24.0	4041	111	3930	11.97
S27	25.0	4025	112	3913	12.34
S28	26.0	4009	113	3896	12.72
S29	27.0	3994	114	3880	13.08
S30	28.0	3979	115	3864	13.44
S31	29.0	3963	115	3848	13.8
S32	29.7	3958	115	3843	13.92
S33	30.0	3916	114	3802	14.83
S34	31.0	3869	114	3755	15.88
S35	32.0	3823	114	3709	16.91
S36	33.0	3815	114	3701	17.09
S37	34.0	3823	113	3710	16.89
S38	35.0	3869	112	3757	15.84
S39	36.0	3916	112	3804	14.79
S40	36.3	3958	112	3846	13.85

S41	37.0	3963	113	3850	13.75
S42	38.0	3979	114	3865	13.41
S43	39.0	3994	114	3880	13.08
S44	40.0	4009	114	3895	12.74
S45	41.0	4025	114	3911	12.38
S46	42.0	4041	115	3926	12.06
S47	43.0	4056	115	3941	11.72
S48	44.0	4072	115	3957	11.36
S49	45.0	4088	115	3973	11.0
S50	46.0	4104	115	3989	10.64
S51	47.0	4120	115	4005	10.29
S52	48.0	4136	115	4021	9.93
S53	49.0	4152	115	4037	9.57
S54	50.0	4168	114	4054	9.18
S55	51.0	4184	114	4070	8.82
S56	52.0	4201	112	4089	8.4
S57	52.8	4204	111	4093	8.32
S58	53.0	4198	109	4089	8.4
S59	54.0	4178	107	4071	8.81
S60	55.0	4158	104	4054	9.18
S61	56.0	4138	102	4036	9.58
S62	57.0	4118	102	4016	10.04
S63	58.0	4098	103	3995	10.51
S64	59.0	4079	106	3973	10.99
S65	60.0	4059	109	3950	11.51
S66	61.0	4040	111	3929	11.99
S67	62.0	4020	112	3908	12.46
S68	63.0	4002	113	3889	12.88
S69	64.0	3982	114	3868	13.34
S70	65.0	3962	114	3848	13.8

S71	66.0	3944	115	3829	14.23
-----	------	------	-----	------	-------

Pré-Esforço Final

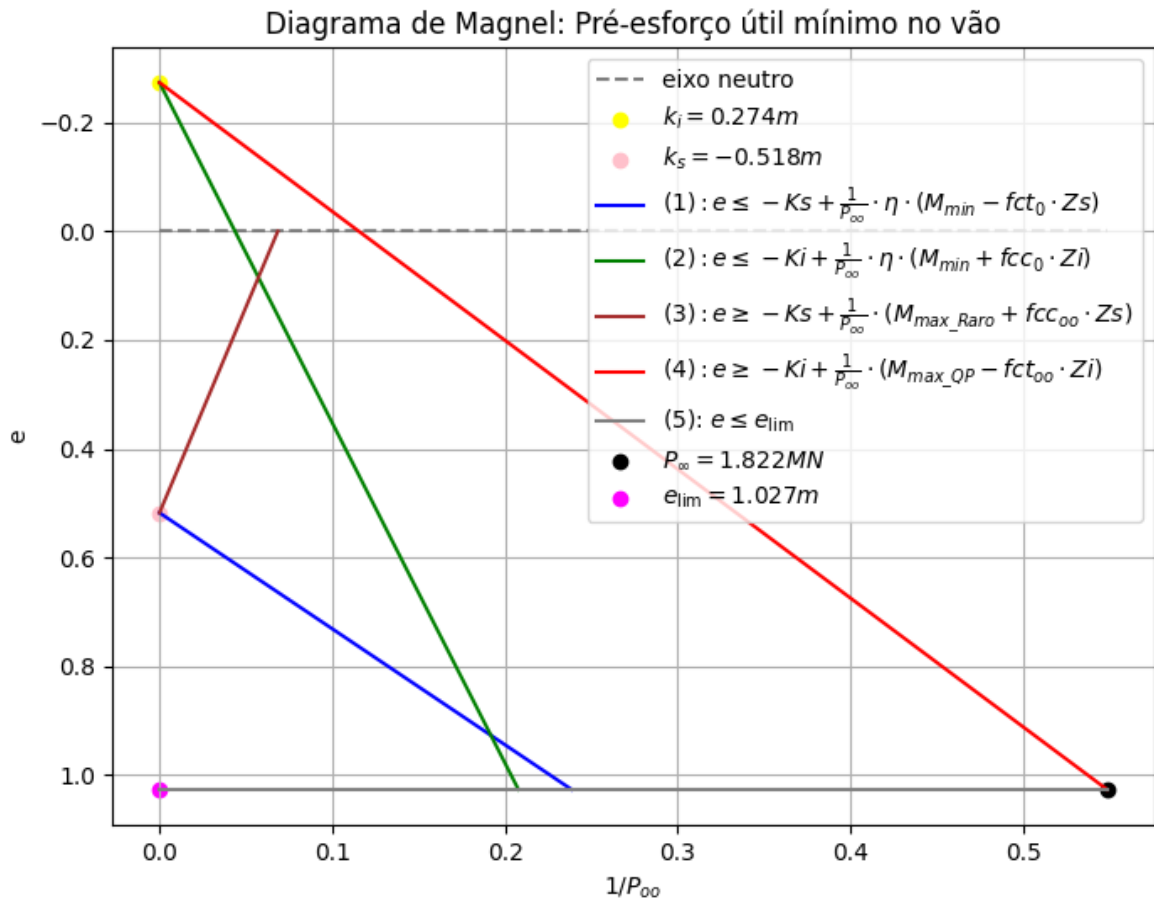
Secção	x [m]	Pm0 [kN]	P_util [kN]
S1	0.0	7830	7600
S2	1.0	7860	7630
S3	2.0	7894	7664
S4	3.0	7928	7698
S5	4.0	7958	7728
S6	5.0	7992	7762
S7	6.0	8024	7796
S8	7.0	8064	7836
S9	8.0	8102	7876
S10	9.0	8137	7913
S11	10.0	8177	7955
S12	11.0	8222	8004
S13	12.0	8262	8050
S14	13.0	8302	8096
S15	13.2	8320	8116
S16	14.0	8314	8110
S17	15.0	8286	8080
S18	16.0	8260	8054
S19	17.0	8228	8022
S20	18.0	8202	7994
S21	19.0	8176	7966
S22	20.0	8155	7943
S23	21.0	8129	7915
S24	22.0	8103	7887
S25	23.0	8077	7857
S26	24.0	8059	7837
S27	25.0	8032	7808

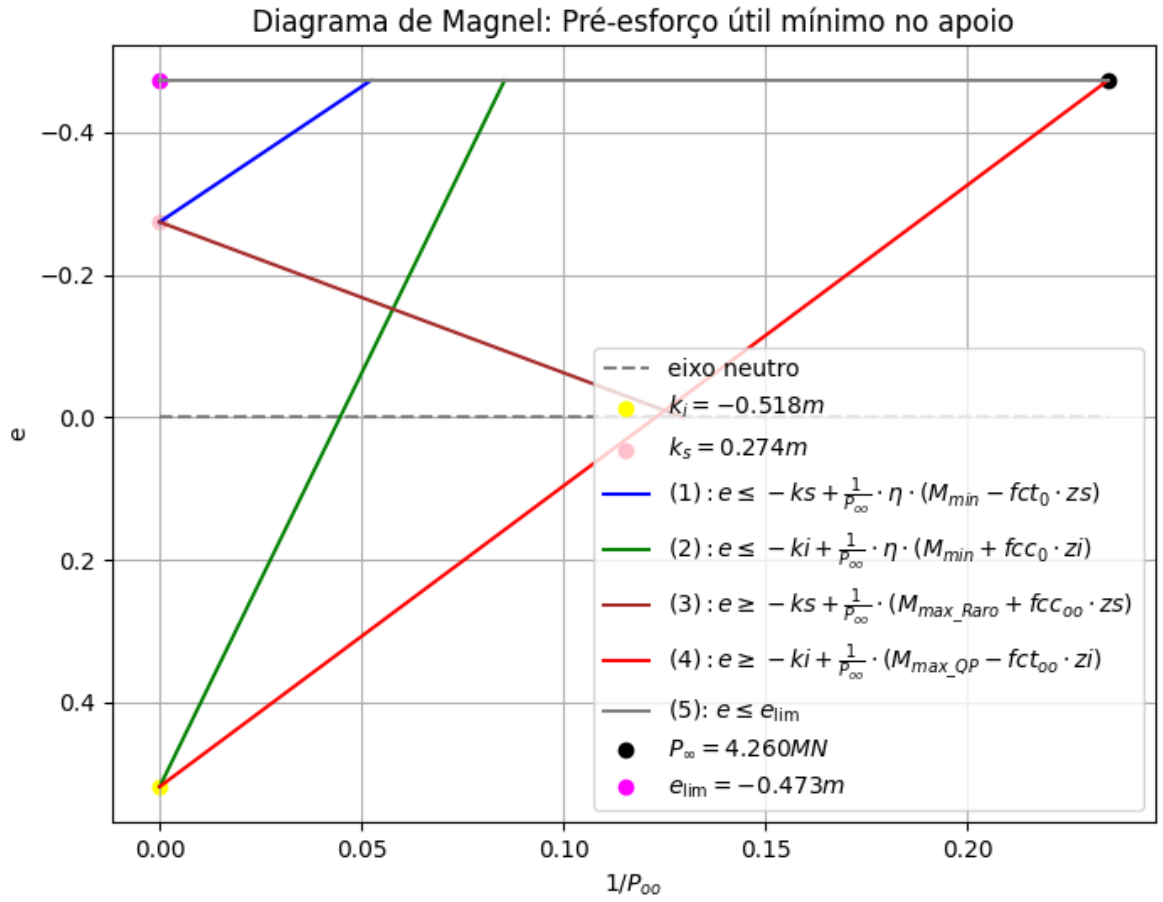
S28	26.0	8012	7786
S29	27.0	7988	7760
S30	28.0	7946	7716
S31	29.0	7908	7678
S32	29.7	7817	7587
S33	30.0	7733	7505
S34	31.0	7644	7416
S35	32.0	7570	7342
S36	33.0	7572	7344
S37	34.0	7576	7350
S38	35.0	7668	7444
S39	36.0	7762	7538
S40	36.3	7852	7628
S41	37.0	7885	7659
S42	38.0	7905	7677
S43	39.0	7930	7702
S44	40.0	7954	7726
S45	41.0	7980	7752
S46	42.0	8000	7770
S47	43.0	8024	7794
S48	44.0	8050	7820
S49	45.0	8077	7847
S50	46.0	8109	7879
S51	47.0	8135	7905
S52	48.0	8161	7931
S53	49.0	8187	7957
S54	50.0	8219	7991
S55	51.0	8245	8017
S56	52.0	8279	8055
S57	52.8	8279	8057

S58	53.0	8267	8049
S59	54.0	8227	8013
S60	55.0	8187	7979
S61	56.0	8153	7949
S62	57.0	8113	7909
S63	58.0	8079	7873
S64	59.0	8047	7835
S65	60.0	8013	7795
S66	61.0	7981	7759
S67	62.0	7946	7722
S68	63.0	7922	7696
S69	64.0	7888	7660
S70	65.0	7860	7632
S71	66.0	7830	7600

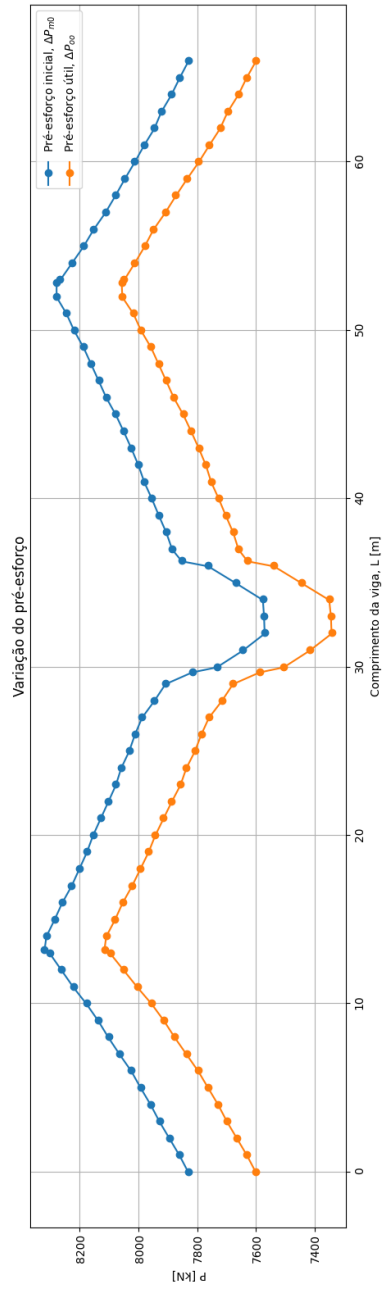
Anexos

Pré-esforço útil mínimo - Vãos





Variação do pré-esforço



Equações que definem o traçado dos cabos

Troço [m]	Equação
1, c1: 13.2	$0.00589657572795403*x^{**2}$
2, c1: 16.5	$1.02741935483871 - 0.00459136822773186*x^{**2}$
3, c1: 3.3	$-0.0229568411386593*x^{**2} - 0.22258064516129$
4, c1: 3.3	$0.0229568411386593*x^{**2} - 0.47258064516129$
5, c1: 16.5	$0.00459136822773186*x^{**2} - 0.22258064516129$
6, c1: 13.2	$0.00589657572795403*x^{**2}$
1, c2: 13.2	$0.00589657572795403*x^{**2}$
2, c2: 16.5	$1.02741935483871 - 0.00459136822773186*x^{**2}$
3, c2: 3.3	$-0.0229568411386593*x^{**2} - 0.22258064516129$
4, c2: 3.3	$0.0229568411386593*x^{**2} - 0.47258064516129$
5, c2: 16.5	$0.00459136822773186*x^{**2} - 0.22258064516129$
6, c2: 13.2	$0.00589657572795403*x^{**2}$

Nota: (**) é o mesmo que (^).

Anexo 2

B. Manual do utilizador

Introdução

O **TendonTech** é um programa de cálculo automático para modelação de cabos para vigas de betão pré-esforçado. Este manual de instruções fornece uma visão geral das funcionalidades do programa, requisitos de sistema, orientações para a utilização e explicações sobre como inserir dados e interpretar os resultados.

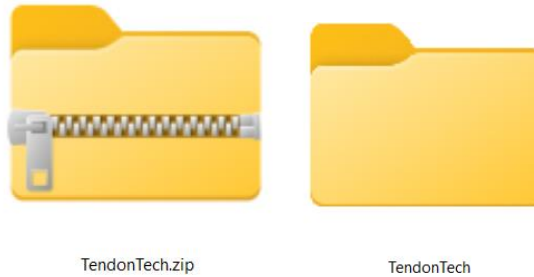
Requisitos de Sistema

O TendonTech é desenvolvido em Python e requer os seguintes requisitos de sistema para funcionar corretamente:

- Sistema operativo: Windows
- [Python==3.8](#) (Windows x86-64 executable installer)
- [Anaconda](#)

Instalação

A instalação do **TendonTech** inicia com o download do ficheiro **TendonTech.zip**. Após o download, recomenda-se que o utilizador extraia os arquivos contidos no ficheiro comprimido para um diretório previamente definido. A definição de um diretório específico facilita a localização e o acesso aos arquivos necessários durante a instalação e utilização do programa.



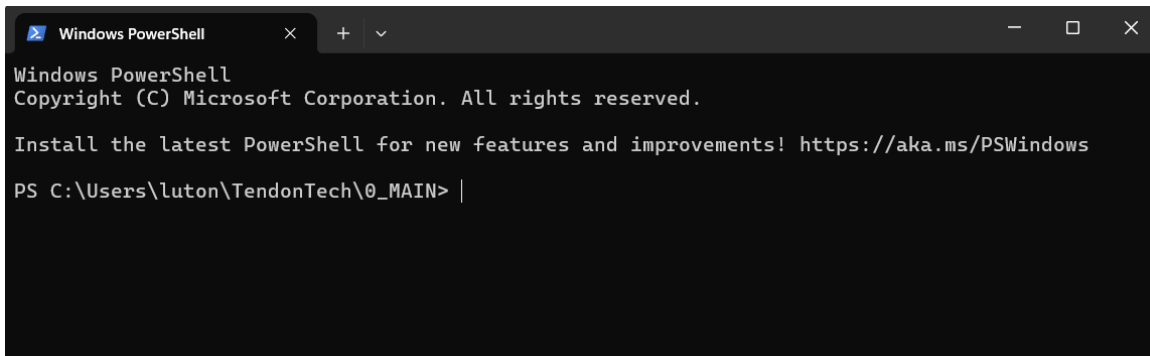
Após a extração, a pasta **TendonTech** será organizada em 8 subpastas, cada uma contendo os componentes necessários para a correta execução e operação do programa.

0_MAIN	Pasta de ficheiros
1_JN	Pasta de ficheiros
2_PLOTS	Pasta de ficheiros
3_FILES	Pasta de ficheiros
4_FILES_DXF	Pasta de ficheiros
5_OUT	Pasta de ficheiros
6_ICON	Pasta de ficheiros
7_INSTRUCTIONS	Pasta de ficheiros

Abra a pasta **0_MAIN** e localize o ficheiro **requirements.txt**. Este ficheiro contém todas as dependências necessárias para garantir o funcionamento adequado do **TendonTech** na sua máquina.

requirements.txt	Documento de Te...
TT_tests.py	JetBrains PyCharm
TendonTech.py	JetBrains PyCharm
TT_BEAM_1.py	JetBrains PyCharm
TT_InputData.py	JetBrains PyCharm
TT_BEAM_2.py	JetBrains PyCharm
TT_BEAM_3.py	JetBrains PyCharm
TT_BEAM_4.py	JetBrains PyCharm
__pycache__	Pasta de ficheiros
.spyproject	Pasta de ficheiros

Ainda na pasta **0_MAIN**, abra o terminal diretamente clicando com o botão direito do mouse e selecionando a opção correspondente para abrir o terminal no local.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

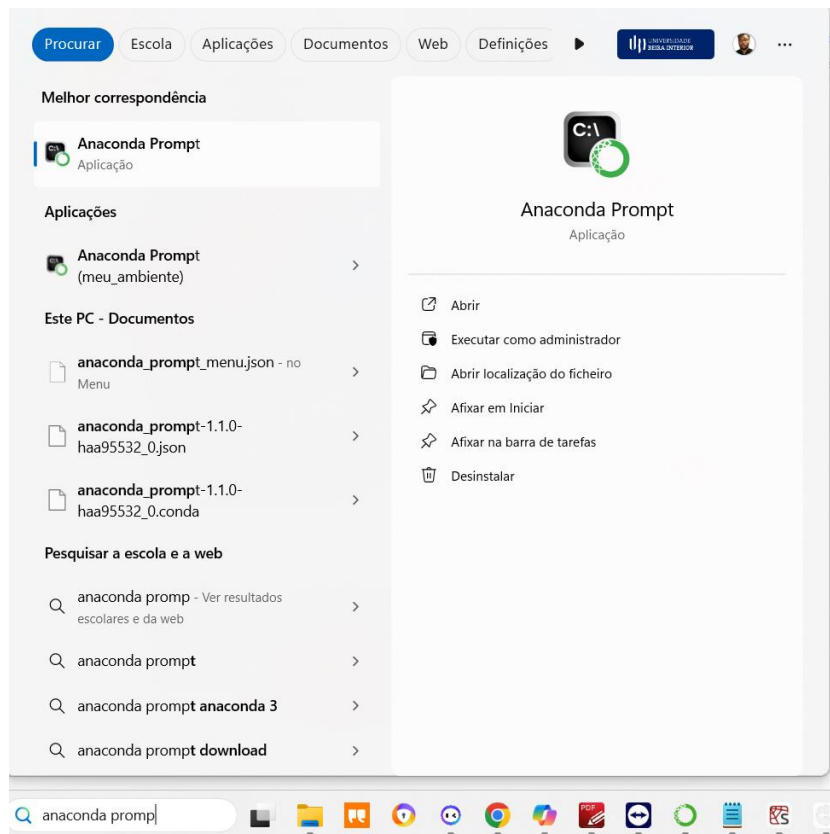
PS C:\Users\luton\TendonTech\0_MAIN> |
```

Na linha de comando, execute os seguintes comandos para configurar o ambiente virtual e instalar as dependências do **TendonTech**:

```
python -m venv venv
source venv/bin/activate # macOS/Linux
venv\Scripts\activate # Windows
pip install -r requirements.txt
```

Após a instalação de todas as dependências, o programa estará configurado e pronto para execução.

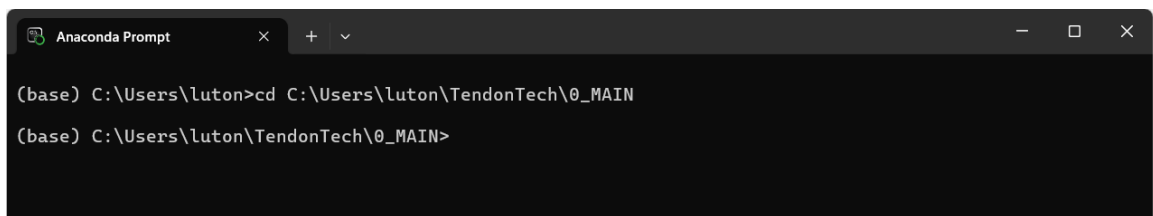
Se a instalação não for concluída com sucesso, o utilizador pode utilizar o **Anaconda**. Para tal, certifique-se de que o **Anaconda Navigator** já está instalado. Em seguida, procure por **Anaconda Prompt** no menu iniciar e execute a aplicação.



Com o **Anaconda Prompt** aberto, insira o seguinte comando:

```
cd C:\Users\luton\TendonTech\0_MAIN
```

Este comando abre o diretório onde se encontra a pasta **0_MAIN**. Note que o caminho para a pasta **0_MAIN** pode variar conforme a organização de diretórios de cada utilizador. Ajuste o caminho conforme necessário.



Em seguida, proceda de forma semelhante à utilização da linha de comando, executando:

```
pip install -r requirements.txt
```

Este comando instalará todas as dependências necessárias para o funcionamento do programa.

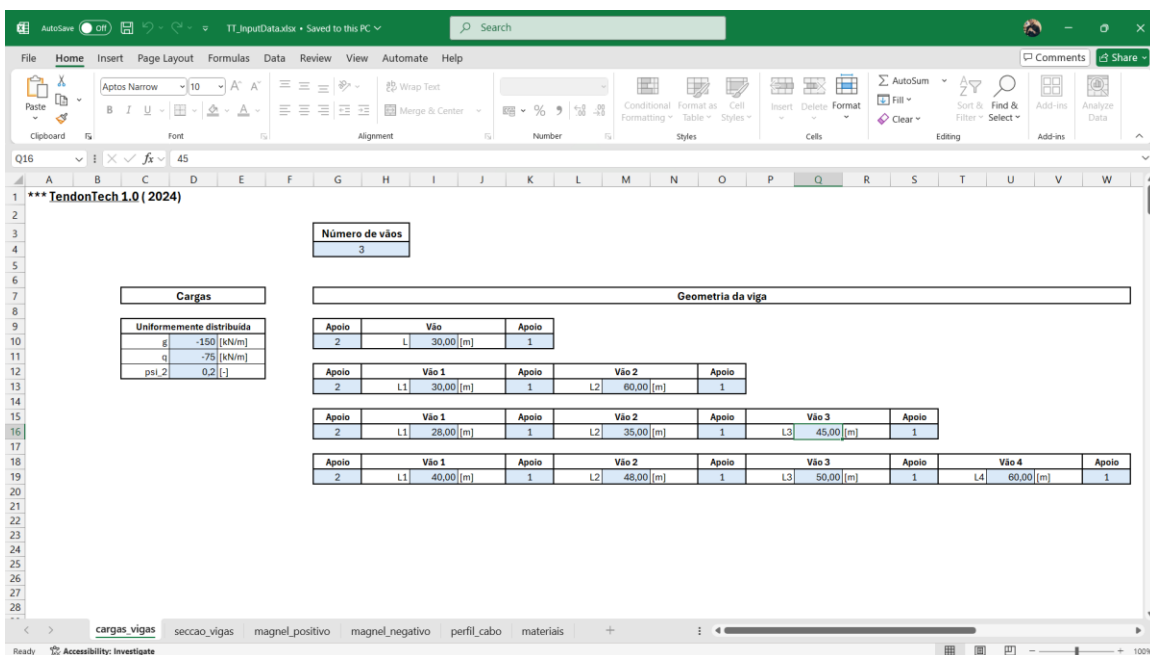
```
Anaconda Prompt
(base) C:\Users\luton>cd C:\Users\luton\TendonTech\0_MAIN
(base) C:\Users\luton\TendonTech\0_MAIN>pip install -r requirements.txt
```

Depois de concluir a instalação de todas as dependências, o programa estará devidamente configurado e pronto para ser executado diretamente através do **Anaconda Prompt**.

Utilização - entrada de dados

Para utilizar o **TendonTech**, o utilizador deverá, primeiramente, preencher o ficheiro **TT_InputData.xlsx**, localizado na pasta **1_IN**.

Este ficheiro contém todos os dados essenciais para a realização dos cálculos. Os campos destacados em azul indicam os dados que podem ser alterados pelo utilizador, permitindo a personalização conforme as necessidades do projeto.



TT_InputData.xlsx - Saved to this PC

File Home Insert Page Layout Formulas Data Review View Automate Help

Clipboard Font Alignment Number Styles Cells Editing Add-ins

G10

*** TendonTech 1.0 (2024)

Seção transversal

Seção I

Seção T

Seção I

Seção Retangular

b	2,600 [m]
s	0,200 [m]
a	1,600 [m]
t	0,450 [m]
d	1,800 [m]

b	3,000 [m]
s	0,800 [m]
h	2,400 [m]
t	1,000 [m]
a	2,400 [m]
d	4,000 [m]

b	1,000 [m]
a	1,300 [m]

Área	A	1,240 [m ²]
------	---	-------------------------

Perímetro	P	8,800 [m]
-----------	---	-----------

Centro Geométrico	cx	0,000 [m]
	cy	1,177 [m]

Momentos de Inércia	Ix	0,400 [m ⁴]
	Iy	0,305 [m ⁴]

Área	A	6,720 [m ²]
------	---	-------------------------

Perímetro	P	16,800 [m]
-----------	---	------------

Centro Geométrico	cx	0,000 [m]
	cy	2,114 [m]

Momentos de Inércia	Ix	12,254 [m ⁴]
	Iy	0,833 [m ⁴]

Área	A	1,300 [m ²]
------	---	-------------------------

Perímetro	P	4,600 [m]
-----------	---	-----------

Centro Geométrico	cx	0,500 [m]
	cy	0,650 [m]

Momentos de Inércia	Ix	0,183 [m ⁴]
	Iy	0,108 [m ⁴]

cargas_vigas secao_vigas magnel_positivo magnel_negativo perfil_cabo materiais

TT_InputData.xlsx - Saved to this PC

File Home Insert Page Layout Formulas Data Review View Automate Help Acrobat

Clipboard Font Alignment Number Styles Cells Editing Add-ins Adobe Ac...

C9

*** TendonTech 1.0 (2024)

Propriedades Geométricas

h	1,800 [m]
Ac	1,240 [m ²]
Centroid z	0,000 [m]
Centroid y	1,177 [m]
Iz	0,400 [m ⁴]
Iy	0,305 [m ⁴]
Izy	0,000 [m ⁴]
1,5_phi_bainha	0,150 [m]
Perímetro	8,800 [m]
vi	1,177 [m]
vs	-0,823 [m]
zi	0,340 [m ³]
zs	-0,842 [m ³]
ki	0,274 [m]
ks	-0,518 [m]
e_lim	1,027 [m]

Material (Betão)

C	30/37
fctd	20 [MPa]
fct_0	0 [MPa]
fcc_0	13,5 [MPa]
fct_oo	0 [MPa]
fcc_oo	18 [MPa]

Material (Aço de pré-esforço)

fpk	1860 [MPa]
A_nominal	1,5 [cm ²]

Material (Betão)

C	30/37
fctd	20 [MPa]
fcm	38 [MPa]
Ecm	33 [GPa]

Fator de correção

eta_Pes/P0	0,9 [-]
------------	---------

cargas_vigas secao_vigas magnel_positivo magnel_negativo perfil_cabo materiais

*** TendonTech 1.0 (2024)
 *** Para a presente versão, apenas Troços parabólicos são suportados

1 Vão

Nº cabos		Vão	
Nº esticamentos por cabo		L	30 [m]
n_cabos	4 [-]		
n_estic	2 [-]		

2 Vãos

Nº cabos		Vão 1		Vão 2		Nº esticar	
Nº esticamentos por cabo		L1	33 [m]	L2	33 [m]	n_cabos	
n_cabos	2 [-]					n_estic	
n_estic	2 [-]						

Cabo 1

Troço	Inicio			Fim			Tipo
	x [m]	y [m]	z [m]	x [m]	y [m]	z [m]	
Troço 1	0	-0,17	-0,15	30	-0,17	-0,15	1
Troço 2	-	-	-	-	-	-	-
Troço 3	-	-	-	-	-	-	-

Cabo 2

Troço	Inicio			Fim			Tipo
	x [m]	y [m]	z [m]	x [m]	y [m]	z [m]	
Troço 1	0	-0,25	0,15	30	-0,25	0,15	1
Troço 2	-	-	-	-	-	-	-
Troço 3	-	-	-	-	-	-	-

Cabo 3

Troço	Inicio			Fim			Tipo
	x [m]	y [m]	z [m]	x [m]	y [m]	z [m]	
Troço 1	0	-0,17	-0,15	30	-0,17	-0,15	1
Troço 2	-	-	-	-	-	-	-
Troço 3	-	-	-	-	-	-	-

Cabo 1

Troço	Inicio			Fim			Tipo
	x [m]	y [m]	z [m]	x [m]	y [m]	z [m]	
Troço 1	0	0	-0,1	13,2	-	-0,1	1
Troço 2	13,2	-	-0,1	29,7	-	-0,1	1
Troço 3	29,7	-	-0,1	33	-	-0,1	1
Troço 4	33	-	-0,1	36,3	-	-0,1	1
Troço 5	36,3	-	-0,1	52,8	-	-0,1	1
Troço 6	52,8	-	-0,1	66	0	-0,1	1
Troço 7	-	-	-	-	-	-	-
Troço 8	-	-	-	-	-	-	-
Troço 9	-	-	-	-	-	-	-
Troço 10	-	-	-	-	-	-	-
Troço 11	-	-	-	-	-	-	-
Troço 12	-	-	-	-	-	-	-

materials

*** TendonTech 1.0 (2024)

Betão

C	30 [MPa]
fcd	20 [MPa]
fct_D	0 [MPa]
fcc_0	13,5 [MPa]
fct_oo	0 [MPa]
fcc_oo	18 [MPa]
fcm	38 [MPa]
Ecm	33 [GPa]

Aço de pré-esforço

Y	1860 [-]
fp01k	1860 [MPa]
fpk	1860,00 [MPa]
Ep	195,00 [GPa]
bainha	100,00 [mm]
A_nominal	1,5 [cm ²]

European Technical Assessment ETA

1, 3, 4, 5, 6, 7, 8, 9, 12, 15, 19, 22, 27, 31, 37, 43, 49, 55

Coefficiente de atrito

mu	0,2 [-]
----	---------

Desvio angular parasita

k	0,010 [rad/m]
---	---------------

Reentrada das cunhas

dstl	8 [mm]
------	--------

Coefficiente de Fluência

t0	7 [dias]
t_oo	50 [anos]
t	18250 [dias]

Ambiente

RH	80 [%]
----	--------

Coefficiente de Retração

fcrmo	10 [MPa]
RHO	100 [%]
Classe do Cimento	N [-]

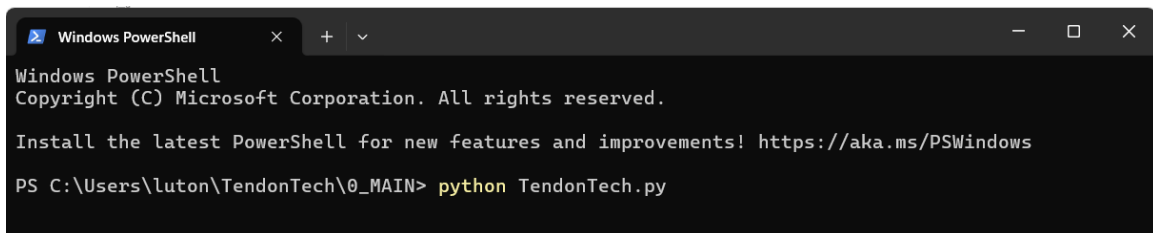
materials

Uma vez o **TT_InputData.xlsx** devidamente preenchido e guardado, o utilizador estará pronto para executar o programa, iniciando o processo de cálculo com os dados fornecidos.

Utilização - TendonTech

Na pasta **0_MAIN**, além dos ficheiros de código adicionais, encontra-se o ficheiro de código principal, **TendonTech.py**, que importa todos os outros componentes necessários para a execução do programa. Para iniciar o processo, abra o terminal diretamente e execute o seguinte comando:

```
python TendonTech.py
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\luton\TendonTech\0_MAIN> python TendonTech.py
```

Se optar por executar o programa através do **Anaconda Prompt**, siga estes passos:

Abra o **Anaconda Prompt**.

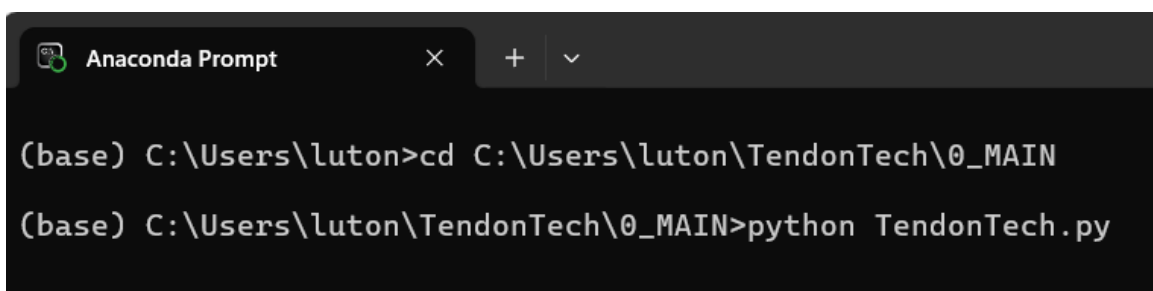
Altere o diretório para a pasta **0_MAIN** com o seguinte comando:

```
cd C:\Users\xxx\TendonTech\0_MAIN
```

Em que 'xxx' é o nome do utilizador.

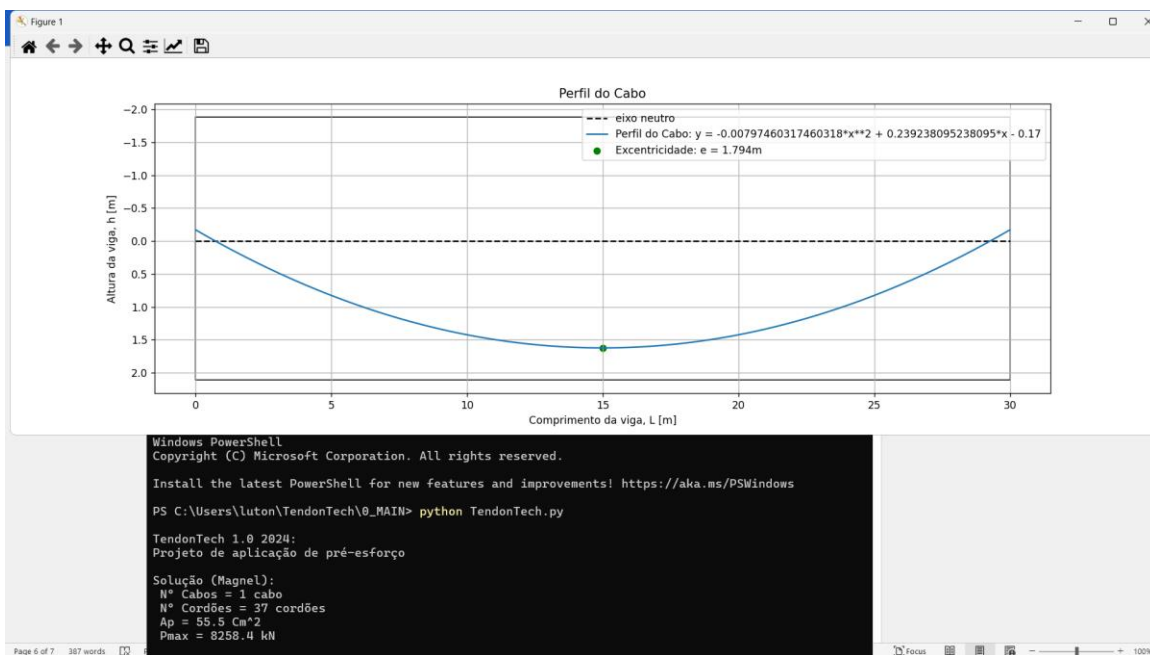
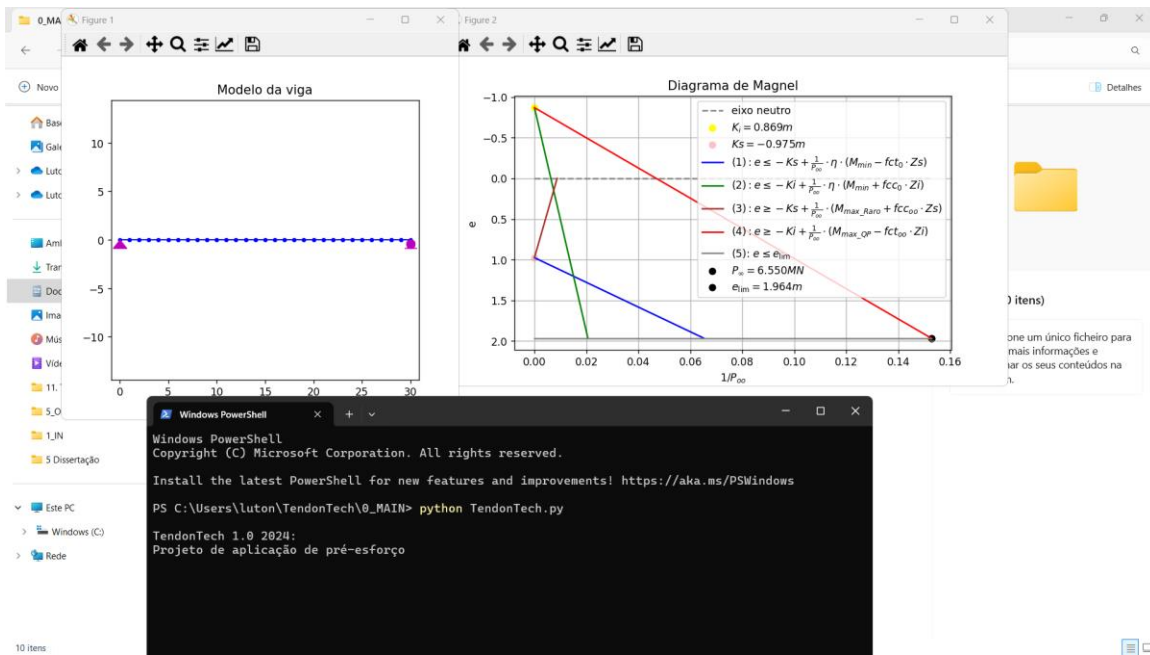
Execute o **TendonTech** utilizando o comando:

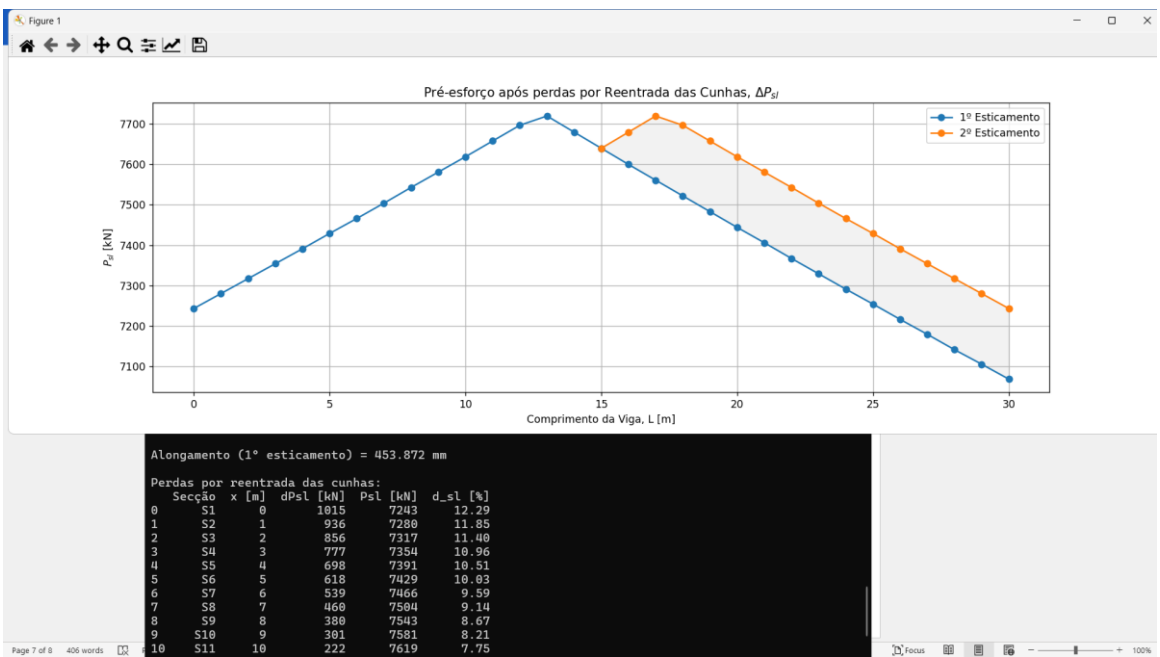
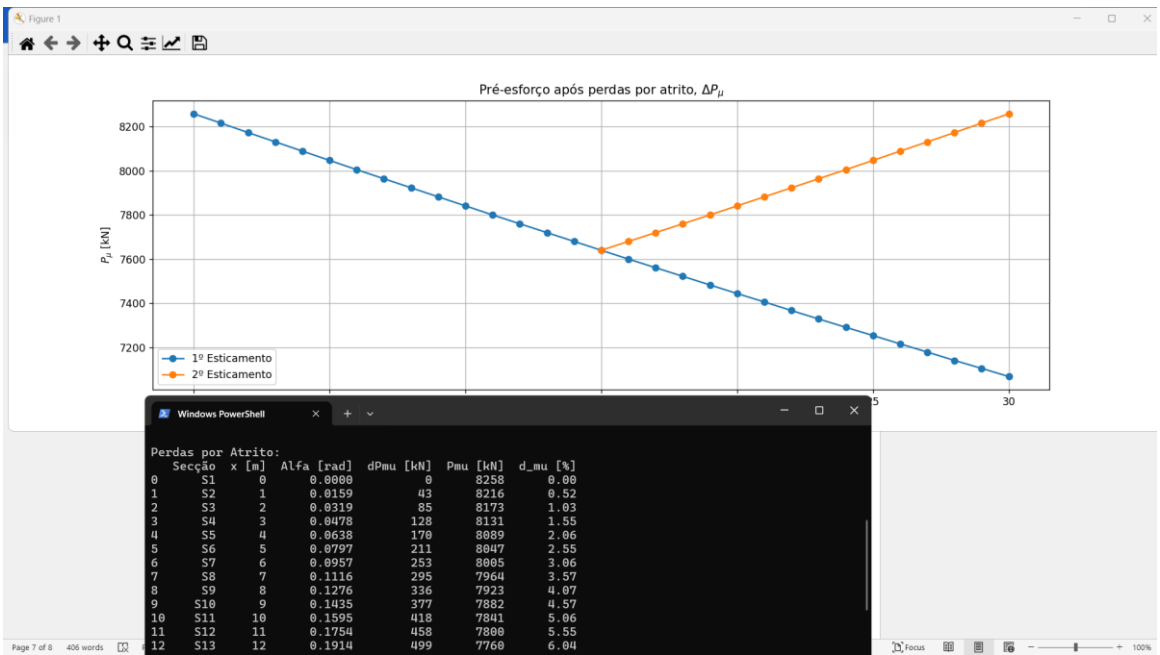
```
python TendonTech.py
```



```
Anaconda Prompt
(base) C:\Users\luton>cd C:\Users\luton\TendonTech\0_MAIN
(base) C:\Users\luton\TendonTech\0_MAIN>python TendonTech.py
```

Os resultados serão apresentados gradualmente, à medida que o cálculo avança em cada fase, seja através da linha de comandos ou do **Anaconda Prompt**. Durante este processo, será necessário fechar os **plots** gerados para que o programa possa continuar com os cálculos subsequentes até à sua conclusão.



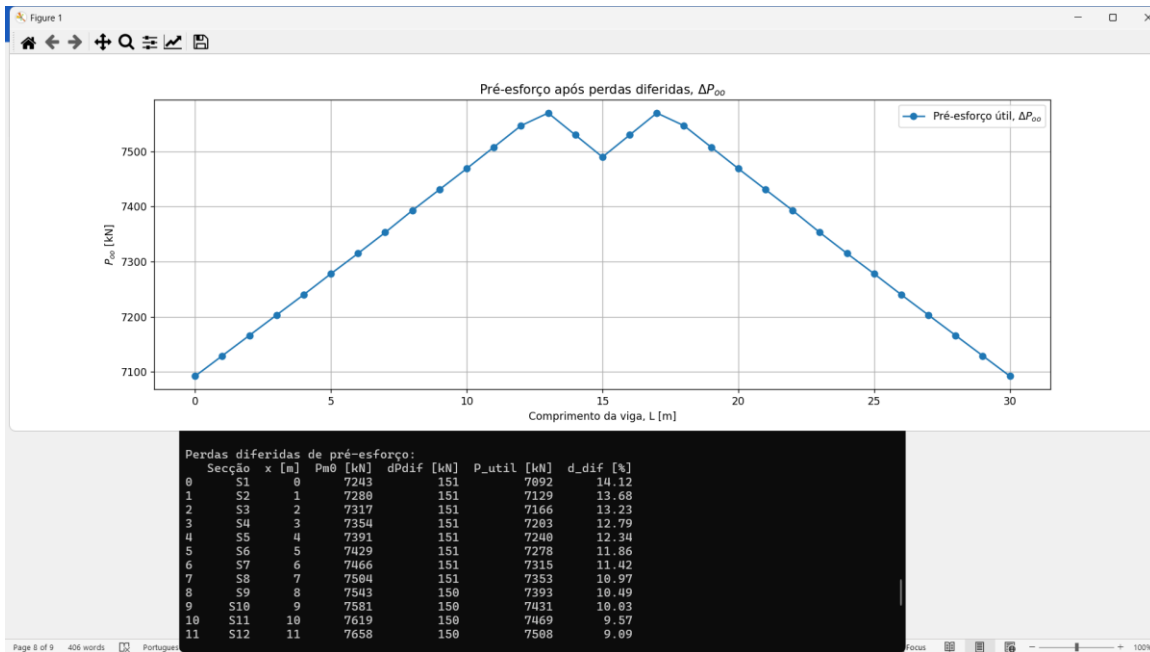


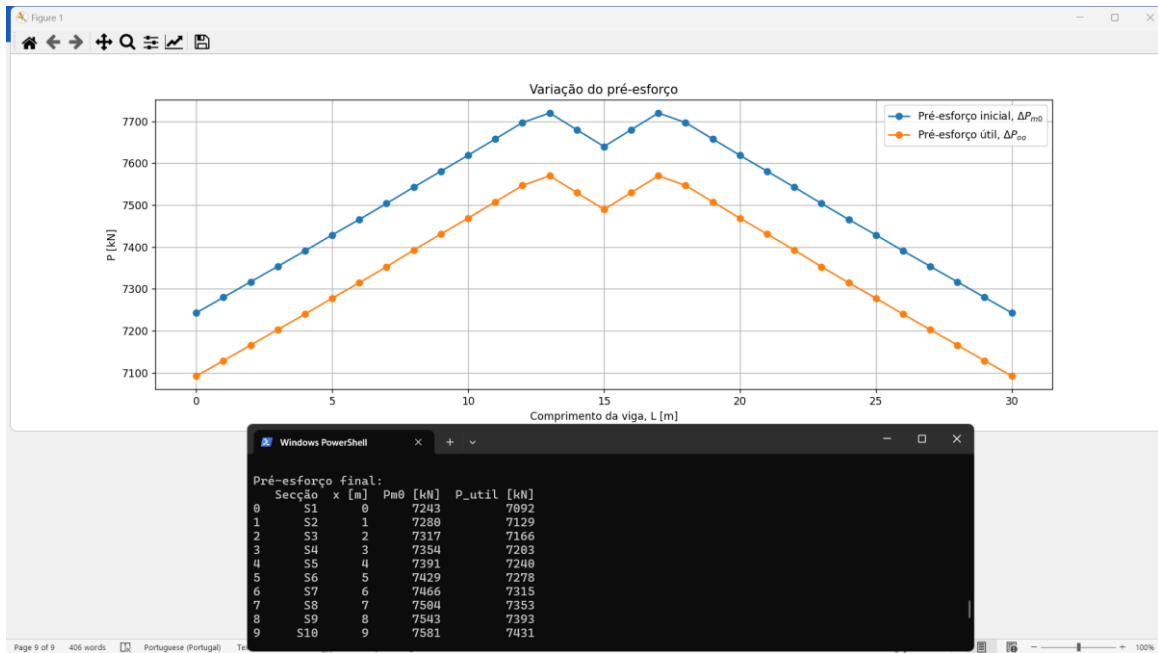
```

Windows PowerShell
Alongamento (2º esticamento) = 50.274 mm

Pré-esforço após Perdas Imediatas:
Seccão x [m] Pm0 [kN] S_cg [MPa] S_cp [MPa] eps_c [mm/m] d_sl [%]
0 S1 0 7243 0.00 -1.08 -0.03 12.29
1 S2 1 7280 0.37 -1.38 -0.03 11.85
2 S3 2 7317 0.72 -1.66 -0.03 11.40
3 S4 3 7354 1.04 -1.93 -0.02 10.96
4 S5 4 7391 1.33 -2.17 -0.02 10.51
5 S6 5 7429 1.60 -2.39 -0.02 10.03
6 S7 6 7466 1.85 -2.59 -0.02 9.59
7 S8 7 7504 2.07 -2.78 -0.02 9.14
8 S9 8 7543 2.26 -2.94 -0.02 8.67
9 S10 9 7581 2.43 -3.08 -0.02 8.21
10 S11 10 7619 2.57 -3.20 -0.02 7.75
11 S12 11 7658 2.68 -3.30 -0.02 7.27
12 S13 12 7697 2.77 -3.38 -0.02 6.80
13 S14 13 7720 2.84 -3.44 -0.02 6.53

```





```

Windows PowerShell
23 S24 23 7504 7353
24 S25 24 7466 7315
25 S26 25 7429 7278
26 S27 26 7391 7240
27 S28 27 7354 7203
28 S29 28 7317 7166
29 S30 29 7280 7129
30 S31 30 7243 7092
100% | 1/1 [00:06<00:00, 6.44s/it]
Calculado !
PS C:\Users\luton\TendonTech\0_MAIN>

```

Após a conclusão dos cálculos, o **TendonTech** gera, na pasta **5_OUT**, um relatório de cálculo nos formatos **.docx** e **.pdf**, contendo os resultados detalhados do processo de cálculo.



TT_CalcReport.docx



TT_CalcReport.pdf

Erros

Caso o programa apresente erros ou problemas durante a execução, siga os passos abaixo para solução:

- Verifique se todos os dados foram inseridos corretamente no ficheiro **TT_InputData.xlsx**.
- Confirme que todas as bibliotecas necessárias estão devidamente instaladas, conforme o ficheiro **requirements.txt**.
- Consulte o log de erros gerado pelo programa para detalhes específicos sobre o problema.
- Caso o problema persista, entre em contato com o desenvolvedor para assistência adicional.

Considerações Finais

O **TendonTech** foi desenvolvido com o objetivo de fornecer uma ferramenta prática e eficiente para o cálculo de cabos em vigas de betão pré-esforçado, simplificando significativamente o processo de modelação e cálculo. Com a automação de tarefas complexas, o programa visa otimizar o tempo e a precisão nos projetos de engenharia de estruturas.