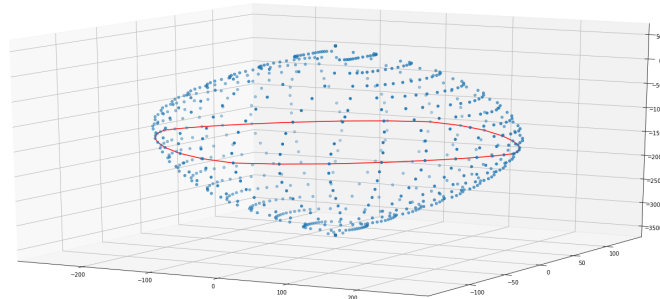




TÉCNICO
LISBOA



Dimensionamento automático de secções de betão armado
Secções em flexão composta desviada submetidas a ações
sísmicas

Andreia Filipa Ferreira Santos

Dissertação para obtenção do Grau de Mestre em

Engenharia Civil

Orientador: Prof. Manuel da Cunha Ritto Corrêa

Júri

Presidente: Prof. Mário Manuel Paisana dos Santos Lopes

Orientador: Prof. Manuel da Cunha Ritto Corrêa

Vogais: Prof. Jorge Manuel Vinagre Alfaiate

Setembro de 2020

Para todos os que tornaram este percurso possível...
Para o meu irmão, uma prova de que tudo é possível com trabalho e dedicação, apesar de todos os
obstáculos, esses mesmos que nos tornam mais fortes...

Agradecimentos

O primeiro agradecimento e muito merecido, sem qualquer dúvida, será para o meu orientador, o professor Manuel Corrêa, pelo seu brilhante desempenho naquilo que é a orientação de uma dissertação. Para além dos seus conhecimentos científicos, quero agradecer especialmente toda a disponibilidade, mesmo nos dias em que essa disponibilidade significava trabalhar fora do horário laboral. O meu muito obrigada!

Um especial agradecimento a quem tornou alcançável todo este caminho: os meus pais. As pessoas importantes na minha vida, os meus pilares desde que me recordo. Vou relembrar cada degrau, cada obstáculo, cada vitória, obrigada pela força e por acreditarem sem hesitação que seria possível.

Para finalizar, um último agradecimento a alguém que se tornou parte de mim e que tornou mais fácil cada pedaço dos meus dias. Ser-te-ei para sempre grata.

Obrigada!

Declaração

Declaro que o presente documento é um trabalho original da minha autoria e que cumpre todos os requisitos do Código de Conduta e Boas Práticas da Universidade de Lisboa.

Resumo

O principal objetivo desta dissertação é desenvolver uma base precisa para o dimensionamento de secções de betão armado sujeitas a diversos tipos de esforços, sendo que a maior atenção é dada para secções sujeitas a flexão composta desviada.

Muitos são os desenvolvimentos em relação a esta matéria, especialmente ao nível da ação sísmica, mas continua a existir uma separação entre a superfície de interação por parte da ação e a superfície resistente.

Para tal, diversos métodos são programados com recurso a uma programação orientada por objetos, neste caso em concreto a programação em *Python*, com o intuito de simultaneamente integrar vários conceitos já existentes e de criar uma base sólida para dimensionar superfícies de resistência da secção transversal de forma mais precisa mantendo os fatores de segurança estabelecidos nos eurocódigos.

Embora seja um problema teórico, ainda que muito abordado pela comunidade científica, continua a carecer de certo tipo de desenvolvimentos que são cada vez mais verosímeis com o aumento exponencial das capacidades computacionais.

Para além do principal foco da dissertação, a modelação tem versatilidade suficiente para tratar vários tipos de secção de forma precisa, bem como dimensionar uma secção com conjuntos de esforços gerais, permitindo resultados tanto numéricos como gráficos, facilitando comparações com valores e tabelas já existentes e continuando a possibilitar o sentido crítico no dimensionamento de estruturas.

Palavras-chave: Secções de betão armado, Superfícies resistentes, Dimensionamento de estruturas, Flexão composta desviada, Interação de esforços

Abstract

The main focus of this dissertation is to develop a precise foundation for the design of reinforced concrete structures subjected to both axial force and bi-axial moments.

There are many developments around this subject, specially when it comes to seismic actions, however a gap remains between the action and the resistance.

With the purpose of diminishing this lacuna, several methods were computed based on an object-oriented programming, in this case programming in *Python*, trying simultaneously integrate various existing concepts and to create a solid foundation for the design of reinforced concrete sections.

Although this is a theoretical problem, even it is studied a lot in the scientific community, there are a few more questions that need to be answered. These might be possible when the increasing computational capabilities is almost exponential.

Beyond the main goal of this dissertation, the program has sufficient versatility to treat many other problems, as design or security check of different kinds of stresses for several types of cross sections, returning the results in a numerical or graphical mode, allowing comparisons with existent tables or graphics, still giving the designer his critical sense.

Keywords: Reinforced concrete sections, Resistant capacity surfaces, Design of structures, Axial force and bi-axial moments, Stress-resultant interaction

Conteúdo

Agradecimentos	v
Declaração	vii
Resumo	ix
Abstract	xi
Lista de Tabelas	xv
Lista de Figuras	xvii
Nomenclatura	xxi
1 Introdução	1
1.1 Estruturação	1
2 Estado de arte	3
3 Secção transversal	7
3.1 Materiais	7
3.2 Equações governativas dos pontos elementares	9
3.2.1 Deformação	9
3.2.2 Tensão	10
3.2.3 Esforços	11
3.2.4 Deformação máxima da secção transversal	12
3.3 Geometria das secções transversais	13
3.3.1 Secção retangular	13
3.3.2 Secção T	15
3.3.3 Secção circular	16
4 Superfícies resistentes de interação	19
4.1 Deformações em estado limite último	19
4.2 Método 1 - Aplicando deformações	21
4.3 Método 2 - Fixando N e θ	21
4.4 Método 3 - Definindo a direção dos esforços	22
4.5 Método 4 - Variando a taxa de armadura	25
4.5.1 Escolha da matriz de rigidez	27

4.6	Validação dos métodos	27
4.6.1	Exemplo 1 - Flexão composta - Secção retangular	27
4.6.2	Exemplo 2 - Flexão composta - Secção circular	31
4.6.3	Exemplo 3 - Flexão composta desviada - Secção retangular	34
5	Esforços resultantes de uma análise sísmica por espectros de resposta	39
5.1	Diagramas de resposta	39
5.2	Extrapolação de métodos para ações sísmicas	40
5.2.1	Método 5	41
5.2.2	Método 6	41
5.2.3	Validação de métodos	42
6	Resultados	47
6.1	Superfícies de interação	47
6.1.1	Secção retangular	47
6.1.2	Secção circular	52
6.1.3	Secção T	56
6.2	Superfícies de resistência para uma análise sísmica	60
6.2.1	Ampliação da componente dinâmica da ação sísmica	60
6.2.2	Otimização da taxa de armadura para uma ação sísmica	65
7	Conclusões	69
	Bibliografia	71
A	Anexo1: Programa principal	A.1

Lista de Tabelas

3.1	Propriedades materiais do betão.	7
3.2	Propriedades materiais do aço.	8
4.1	Secção transversal: exemplo 1.	27
4.2	Esforços de dimensionamento: exemplo 1.	28
4.3	Taxa de armadura ω pelos ábacos de betão: exemplo 1.	29
4.4	Resultados para o método 4: exemplo 1.	29
4.5	Resultados para o método 3: exemplo 1.	30
4.6	Resultados para o método 2: exemplo 1.	30
4.7	Resultados para o método 1: exemplo 1.	31
4.8	Secção transversal: exemplo 2.	31
4.9	Esforços de dimensionamento: exemplo 2.	31
4.10	Taxa de armadura ω pelos ábacos de betão: exemplo 2.	32
4.11	Resultados para o método 4: exemplo 2.	33
4.12	Resultados para o método 3: exemplo 2.	33
4.13	Resultados para o método 2: exemplo 2.	34
4.14	Resultados para o método 1: exemplo 2.	34
4.15	Secção transversal: exemplo 3.	34
4.16	Esforços de dimensionamento: exemplo 3.	35
4.17	Taxa de armadura ω pelos ábacos de betão: exemplo 3.	36
4.18	Resultados para o método 4: exemplo 3.	36
4.19	Resultados para o método 3: exemplo 3.	36
4.20	Resultados para o método 1: exemplo 3.	37
5.1	Secção transversal: exemplo.	42
6.1	Secção retangular.	47
6.2	Secção circular.	52
6.3	Secção T.	56
6.4	Secção retangular.	60

Lista de Figuras

3.1	Diagrama tensão-extensão simplificado do betão.	8
3.2	Diagrama tensão-extensão do aço.	9
3.3	Diagrama de extensões para compressão pura.	12
3.4	Secção retangular.	13
3.5	Subdivisão da secção retangular em áreas elementares.	14
3.6	Centro gravítico dos retângulos elementares para a secção retangular.	14
3.7	Posicionamento dos varões de aço para a secção retangular.	15
3.8	Secção T.	15
3.9	Secção $T = R1 + R2$	16
3.10	Centro gravítico da secção T.	16
3.11	Posicionamento dos varões de aço para a secção T.	16
3.12	Secção circular.	17
3.13	Subdivisão da secção circular.	17
3.14	Posicionamento dos varões de aço para a secção circular.	17
4.1	<i>Script</i> da secção transversal - exemplo 1.	28
4.2	Secção transversal - exemplo 1.	28
4.3	<i>Script</i> do método 4 - exemplo 1.	29
4.4	<i>Script</i> do método 3 - exemplo 1.	30
4.5	<i>Script</i> do método 2 - exemplo 1.	30
4.6	<i>Script</i> da secção transversal - exemplo 2.	31
4.7	Secção transversal - exemplo 2.	32
4.8	<i>Script</i> do método 4 - exemplo 2.	33
4.9	<i>Script</i> do método 3 - exemplo 2.	33
4.10	<i>Script</i> do método 2 - exemplo 2.	34
4.11	<i>Script</i> da secção transversal - exemplo 3.	35
4.12	Secção transversal - exemplo 3.	35
4.13	<i>Script</i> do método 4 - exemplo 3.	36
4.14	<i>Script</i> do método 3 - exemplo 3.	36
5.1	Superfície de interação da resposta sísmica - relação entre os vetores v e t	41

5.2	Secção transversal - exemplo para validação.	42
5.3	Superfície de interação da ação sísmica - validação método 5.	42
5.4	Sobreposição $SI-SR_{inicial}$ - exemplo de validação.	43
5.5	<i>Script</i> - validação método 5.	43
5.6	Sobreposição $SI - SR_{inicial} - SR_{ótima}$ - exemplo de validação.	43
5.7	Sobreposição $SI-SR_{inicial}$ - exemplo de validação.	44
5.8	<i>Script</i> - validação método 6.	44
5.9	Sobreposição $SI - SR_{inicial} - SR_{ótima}$ - exemplo de validação.	45
6.1	<i>Script</i> para modulação de curvas de interação - Secção retangular.	48
6.2	Secção retangular - discretização.	48
6.3	Curva de interação $N - M_y$ - Secção retangular.	48
6.4	Curva de interação - perspetiva - $N - M_y - M_z$ - Secção retangular.	49
6.5	Curva de interação - projeção $M_y - M_z$ - Secção retangular.	49
6.6	Curva de interação - projeção $N - M_y$ - Secção retangular.	50
6.7	Curva de interação - projeção $N - M_z$ - Secção retangular.	50
6.8	<i>Script</i> para modulação de curvas de interação - Secção circular.	52
6.9	Secção circular - discretização.	52
6.10	Curva de interação $N - M_y$ - Secção circular.	53
6.11	Curva de interação - perspetiva - $N - M_y - M_z$ - Secção circular.	53
6.12	Curva de interação - projeção $M_y - M_z$ - Secção circular.	54
6.13	Curva de interação - projeção $N - M_y$ - Secção circular.	54
6.14	Curva de interação - projeção $N - M_z$ - Secção circular.	55
6.15	<i>Script</i> para modulação de curvas de interação - Secção T.	56
6.16	Secção T - discretização.	56
6.17	Curva de interação $N - M_y$ - Secção T.	57
6.18	Curva de interação - perspetiva - $N - M_y - M_z$ - Secção T.	58
6.19	Curva de interação - projeção $M_y - M_z$ - Secção T.	58
6.20	Curva de interação - projeção $N - M_y$ - Secção T.	59
6.21	Curva de interação - projeção $N - M_z$ - Secção T.	59
6.22	<i>Script</i> da secção transversal retangular.	60
6.23	Secção transversal retangular.	60
6.24	<i>Script</i> para o método 5 - Verificação de segurança.	61
6.25	Superfícies de interação e resistência - Perspetiva 3D (1).	61
6.26	Superfícies de interação e resistência - Perspetiva 3D (2).	62
6.27	Superfícies de interação e resistência - Perspetiva 3D (3).	62
6.28	Superfícies de interação e resistência - Perspetiva 3D (4).	63
6.29	Superfícies de interação e resistência - Perspetiva 3D (5).	63
6.30	Superfície de interação e resistência - Projeção 2D.	64

6.31 Superfície de interação final e resistência - Projeção 2D.	64
6.32 <i>Script</i> para método 6 - Verificação de segurança.	65
6.33 Superfícies de interação e resistência - Perspetiva 3D (1).	65
6.34 Superfícies de interação e resistência - Perspetiva 3D (2).	66
6.35 Superfícies de interação e resistência - Perspetiva 3D (3).	66
6.36 Superfícies de interação e resistência - Perspetiva 3D (4).	67
6.37 Superfícies de interação e resistência - Perspetiva 3D (5).	67
6.38 Superfície de interação e resistência final - Projeção 2D.	68

Nomenclatura

Variáveis gregas

α	Fator de redução para o valor de rotura do betão à compressão.
χ	Curvatura.
λ	Fator multiplicativo para o método 3.
μ	Momento fletor reduzido.
ν	Esforço axial reduzido.
ω	Taxa de armadura.
ϕ	Diâmetro dos varões de aço.
ρ_{ij}	Coefficientes de correlação modal para a ação sísmica.
σ_c	Tensão do betão.
σ_s	Tensão do aço.
θ	Ângulo de variação nas iterações.
ε	Extensão.
ε_G	Extensão no centro gravítico da secção transversal.
ε_{c2}	Extensão de cedência do betão.
ε_{cu2}	Extensão última de dimensionamento do betão.
ε_{ud}	Extensão última convencional de dimensionamento do aço.
ε_{uk}	Extensão última característica do aço.
ε_{uk}	Valor característico da extensão última do aço.
ε_{yd}	Extensão de cedência do aço.

Variáveis latinas

A	Área.
-----	-------

b	Largura da secção retangular.
b_f	Largura do banzo da secção T.
d	Diâmetro da secção circular.
E_c	Módulo de elasticidade do betão.
E_s	Módulo de elasticidade do aço.
E_t	Tangente do módulo de elasticidade.
f_ω	Fator multiplicativo para o método 4.
f_{cd}	Valor de cálculo de rotura do betão à compressão.
f_{ck}	Valor característico de rotura do betão à compressão.
f_{yd}	Valor de cálculo de cedência do aço.
f_{yk}	Valor característico de cedência do aço.
h	Altura da secção retangular.
h_w	Altura da alma da secção T.
M	Momento fletor.
N	Esforço axial.
n_g	Número de componentes de base para a ação sísmica.
n_y	Número de subdivisões da altura da secção retangular.
n_z	Número de subdivisões da largura da secção retangular.
t_f	Espessura do banzo da secção T.
t_w	Espessura da alma da secção T.
c	Recobrimento da secção transversal.
n	Número de modos para a ação sísmica.

Subscritos

i, j	Índices matriciais.
x, y, z	Coordenadas cartesianas.

Sobrescritos

K	Matriz de flexibilidade.
S	Matriz de correlação para a ação sísmica.

- s Esforços atuantes de uma ação sísmica.
- t Vetor dos fatores da combinação linear para a ação sísmica.
- T Matriz transposta.

Capítulo 1

Introdução

A presente dissertação tem como principal objetivo responder de forma concisa a uma falha que se tem tentado diminuir na relação entre os esforços atuantes e os esforços resistentes, dando especial ênfase às combinações de esforços de flexão composta desviada, sendo uma das principais raízes a ação sísmica.

Vários estudos efetuados no âmbito das superfícies de interação de uma ação sísmica conferem uma forma elipsoidal para a mesma, baseada numa análise por espetros de resposta. A não quantificação da correlação entre os esforços de dimensionamento e a combinação de esforços máximos, que podem não ocorrer simultaneamente, levam ao sobredimensionamento das secções transversais.

Estas afirmações criam uma dificuldade acrescida no dimensionamento de uma secção uma vez que a otimização da mesma só é possível através de processos iterativos. Com os avanços tecnológicos e científicos crê-se que é possível criar metodologias que proporcionem uma secção mais otimizada, e que possam ser mais competitivas do ponto de vista económico, sem que por isso deixem de ser seguras estruturalmente.

1.1 Estruturação

A dissertação é estruturada em sete capítulos, com os respetivos conteúdos descritos sumariamente:

Capítulo segundo - Estado de arte: Pequeno resumo das metodologias existentes e desenvolvidas ao longo do tempo, com especial atenção para as superfícies interativas das ações sísmicas.

Capítulo terceiro - Secção transversal: Tal como o nome indica, o capítulo apresenta os diferentes tipos de secção que podem ser utilizados, ou qualquer composição dos mesmos, integrando os materiais e as suas propriedades através dos pontos elementares materiais pelos quais a secção é dividida. É também neste capítulo que as equações governativas da secção são apresentadas, obtidas a partir das equações relativas aos pontos materiais.

Capítulo quarto - Superfícies resistentes de interação: As metodologias criadas para determinar um ponto da superfície de resistência são explicadas e validadas aqui, sendo que os dois primeiros métodos servem de base para os seguintes e, por consequência, os métodos 3 e 4 serão os utilizados nos seguintes capítulos.

Capítulo quinto - Esforços resultantes de uma análise sísmica por espectros de resposta: Definidos os métodos para definir um ponto da superfície de resistência da secção transversal, é necessário extrapolar o problema para uma superfície. Os métodos 5 e 6 são criados com base nos métodos anteriores, para resolver esta questão.

Capítulo sexto - Resultados: Os resultados são divididos em 2 secções, sendo que a primeira tem o objetivo de criar curvas e/ou superfícies resistentes com uma dada taxa de armadura de uma forma eficaz para vários tipos de secção transversal, e a segunda secção define o fator de segurança de uma ação sísmica relativamente à superfície de resistência da secção transversal bem como a taxa de armadura mínima de uma secção transversal para resistir às ações sísmicas caso esta se mantenha.

Capítulo sétimo - Conclusões: As conclusões resumem os resultados obtidos, reforçando a necessidade de olhar para este problema do lado da resistência das secções. Contudo, apesar de serem necessários maiores desenvolvimentos face a esta questão, este poderá ser o caminho a seguir.

Capítulo 2

Estado de arte

Com a presente evolução construtiva, provocada por um aumento populacional quase exponencial, a engenharia civil tem um papel fundamental no que concerne à otimização de soluções estruturais competitivas de um ponto de vista económico sem que possam comprometer a segurança, mantendo também o conforto. Um relevante assunto a este respeito que tem sido estudado ao longo dos tempos é a ação sísmica. O Eurocódigo 8 [2] recomenda o uso dos espectros de resposta para o dimensionamento das estruturas. No entanto, uma análise que não tenha em consideração todas as correlações pode sobredimensionar as mesmas.

Os primeiros desenvolvimentos acerca deste assunto foram explorados por Gupta (1977) [16]: *Design of column sections subjected to three components of earthquake*. Gupta conseguiu provar que o diagrama de interação para um sismo usando uma combinação quadrática toma a forma de um elipsoide. Tendo em consideração que o elipsoide tem uma forma analítica bem conhecida, foi um grande avanço para o desenvolvimento do tema. Na prática, para que a secção transversal resista às ações sísmicas, correspondentes ao elipsoide, será somente necessário obter um diagrama resistente que envolva completamente o mesmo. Para otimizar a secção utilizando a mínima quantidade de armadura, será esta cláusula suficiente, o que não é o caso das situações correntes para o dimensionamento de estruturas, como já referido pelo EC8, o que pode levar a um sobredimensionamento da estrutura e, o que para uma construção relativamente corrente e de volumes inferiores possa não ter uma preponderância significativa, terá certamente em estruturas complexas e/ou de maior dimensão. Por este motivo, faz sentido abordar o tema mais exhaustivamente, especialmente do lado da resistência, ou seja, no dimensionamento da secção.

Menun e Kiureghian (2000) [7], em *Envelopes for Seismic Response Vectors*., afirmam que no dimensionamento de uma estrutura sujeita a ações sísmicas, os efeitos simultâneos das forças a atuar na estrutura devem ser considerados. A utilização, no seu dimensionamento, do espectro de resposta, identifica os valores máximos das ações a atuar separadamente, sem que tenha em conta que estas poderão nunca ocorrer simultaneamente. Menun e Kiureghian afirmam que, para uma orientação conhecida dos eixos principais, a superfície de interação tem a forma de um elipsoide. Já quando esta

orientação é desconhecida, é definido um *supreme envelope*, construído a partir da orientação crítica dos eixos principais. Sobrepondo uma superfície resistente que envolva completamente o diagrama de interação, obtêm-se resultados muito eficazes para a otimização da estrutura.

Explorando os desenvolvimentos anteriores, e com o intuito de correlacionar a ação sísmica com a resistência da secção de uma forma mais expedita, Rosati e colegas (2008) [14], em *Enhanced solution strategies for the ultimate strength analysis of composite steel-concrete sections subject to axial force and biaxial bending*, desenvolvem vários algoritmos para verificar se a envolvente da ação sísmica, o elipsoide, está completamente contida no domínio da secção resistente. Tendo por objetivo atingir a resistência última da secção sem que necessite de criar toda a superfície, baseia-se numa convergência da secante da matriz de rigidez, de forma a reduzir o número de iterações, visto um dos principais problemas ser a quantidade de equações requeridas para a criação eficiente de uma superfície de resistência otimizada.

Em 2014, Erlicher [3]: *Seismic design by the response spectrum method: A new interpretation of elliptical response envelopes and a novel equivalent static method based on probable linear combinations of modes*, reforça o facto de o dimensionamento ser, em muitas ocasiões, baseado numa resposta conservativa, já que o espectro de resposta providencia o valor máximo da ação para uma direção de forma independente, sem tomar em consideração a probabilidade de simultaneidade das respostas nas diferentes direções do sismo. Neste caso, o diagrama de interação corresponderia a um paralelepípedo. Considerando que estas respostas não ocorrem, geralmente, em simultâneo, Erlicher acredita que o chamado *hyper-ellipsoid*, considerando as probabilidades de ocorrência das diferentes respostas no tempo, é o caminho a seguir. Para tal, formula equações para a criação de um poliedro que se aproxima da superfície de interação. Propõe também uma nova abordagem para o método das forças equivalentes, sendo que o principal problema, considera, é a dificuldade em definir um campo de forças estáticas que sejam representativas de um comportamento não linear.

Sessa e colegas [15], em 2015, com o artigo *Effective use of seismic response envelopes for reinforced concrete structures*, continua a desenvolver métodos para o dimensionamento das secções resistentes, com uma abordagem na qual intitula de *Seismic critical multiplier*, alertando novamente para o facto de o EC8 não ter em consideração as direções provenientes do sismo que serão, à partida, desconhecidas. Este conceito tem em conta novas pesquisas ocorridas no campo das ações sísmicas, com um *Supreme Envelope* [7, 8] com a combinação CQC3 [17], que envolve as várias ações sísmicas independentemente do seu ângulo de ocorrência ou as suas alterações no tempo. O objetivo será majorar a superfície do envelope, até que encontre um ponto tangente à superfície resistente e esta envolva completamente o envelope. Esse multiplicador, já referido, é o fator de segurança da secção resistente em relação à ação exercida.

Uma abordagem para a construção dos diagramas de interação, em [13] (2020): *The shape of the seismic response interaction diagram: The effect of combination rules in response spectrum analysis*, estuda diferentes combinações modais e de componentes da ação sísmica para a sua construção. Concluem que o diagrama de interação pode ser um poliedro, um elipsóide ou algo intermédio consoante as regras adotadas. Contudo, esta abordagem não se foca na resistência da secção, salientando uma

vez mais a pesquisa em torno da ação solicitada.

Continua a existir algum distanciamento entre o cálculo da ação exercida e o da resistência da secção. Esta dissertação foca-se quase exclusivamente na construção das superfícies de interação resistentes, tentando criar ferramentas para preencher esta lacuna.

Capítulo 3

Secção transversal

O capítulo introdutório aborda a modelação das secções transversais comumente usadas na construção civil, sendo elas retangulares, circulares, ou secções T. No entanto, é possível criar uma secção composta combinando qualquer uma destas secções. Cada secção transversal é composta por um conjunto de pontos materiais elementares nos quais a secção é discretizada. Têm, por esse motivo, uma quantidade de atributos, tais como a área elementar e as coordenadas do ponto, e sendo definidos inicialmente, podem ser tão refinados quanto necessário, dependendo do grau de precisão que se queira obter.

3.1 Materiais

As classes do betão e do aço são providenciadas pelo projetista, uma vez que o programa possibilita a sua escolha. As classes possíveis e as suas respetivas propriedades estão disponíveis nas Tabelas 3.1 e 3.2, respetivamente. Note-se que o valor $\varepsilon_{ud} = 0.9 \times \varepsilon_{uk} = 10\%$, na Tabela 3.2, é um valor de referência recomendado pelo REBAP [6] para a limitação da extensão última de dimensionamento do aço. No entanto, os valores característicos ε_{uk} dependem da classe de ductilidade do aço, e estes são da ordem dos 25 a 75% [1]. Como aos 10% de extensão o aço já se encontra em cedência, as diferenças na avaliação das capacidades resistentes à flexão não são muito significativas, razão pela qual se manteve o valor recomendado de 10%.

Tabela 3.1: Propriedades materiais do betão

	C12/15	C15/20	C20/25	C25/30	C30/37	C35/45	C40/50	C45/55	C50/60	
f_{ck}	12.0	15.0	20.0	25.0	30.0	35.0	40.0	45.0	50.0	MPa
f_{cd}	8.0	10.0	13.3	16.7	20.0	23.3	26.7	30.0	33.3	MPa
ε_{c2}	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	‰
ε_{cu2}	-3.5	-3.5	-3.5	-3.5	-3.5	-3.5	-3.5	-3.5	-3.5	‰
E_c	27.0	29.0	30.0	31.0	33.0	34.0	35.0	36.0	37.0	GPa

O diagrama da relação constitutiva tensão-extensão simplificado do betão toma a forma de uma

Tabela 3.2: Propriedades materiais do aço

	A235	A400	A500	
f_{yk}	235.0	400.0	500.0	MPa
f_{yd}	204.3	347.8	434.8	MPa
ε_{yd}	1.02	1.74	2.18	%
ε_{ud}	10.0	10.0	10.0	%
E_s	200	200	200	GPa

parábola e de um retângulo, representado na Equação (3.1) e na Figura 3.1.

$$\sigma_c(\varepsilon_c) = \begin{cases} \alpha f_{cd} & , \varepsilon_c < \varepsilon_{c2} \\ \alpha f_{cd} [1 - (1 - \frac{\varepsilon_c}{\varepsilon_{c2}})^2] & , 0 > \varepsilon_c \geq \varepsilon_{c2} \\ 0 & , \varepsilon_c \geq 0 \end{cases} \quad (3.1)$$

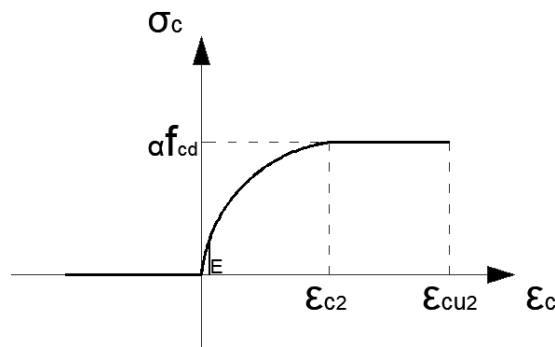


Figura 3.1: Diagrama tensão-extensão simplificado do betão

O coeficiente α , na multiplicação por f_{cd} , considera a diminuição da tensão de rotura do betão quando este é sujeito a tensões elevadas durante um longo período de tempo. O coeficiente pode variar entre 0.85 e 1.00. O EC8 recomenda $\alpha = 1.00$ para situações correntes, ainda que com carregamentos prolongados, sendo que nestas o betão estará solicitado a níveis de tensão relativamente inferiores aos de rotura. Este coeficiente pode ser definido pelo utilizador, sendo por defeito igual a 1.00.

Note que, apesar da Tabela 3.1 apresentar o valor do módulo de elasticidade do betão em regime elástico, é importante considerar a sua rigidez correlacionada com o diagrama parábola-retângulo. A sua forma, em (3.2), passa a receber o nome de Elasticidade tangente, E_t , para cada um dos materiais.

$$E_t(\varepsilon_c) = \begin{cases} 0 & , \varepsilon_c < \varepsilon_{c2} \\ 2\alpha f_{cd} \frac{\varepsilon_{c2} - \varepsilon_c}{\varepsilon_{c2}^2} & , 0 > \varepsilon_c \geq \varepsilon_{c2} \\ 0 & , \varepsilon_c \geq 0 \end{cases} \quad (3.2)$$

No caso do aço, considerar-se-á uma relação constitutiva tensão-extensão elasto-plástica, como apresenta a Equação (3.4) e a Figura 3.2, não considerando o endurecimento do aço após cedência.

$$\sigma_s(\varepsilon_s) = \begin{cases} \frac{\varepsilon_s}{|\varepsilon_s|} f_{yd} & , |\varepsilon_s| > \frac{f_{yd}}{E_s} \\ \varepsilon_s E_s & , \text{caso contrário} \end{cases} \quad (3.3)$$

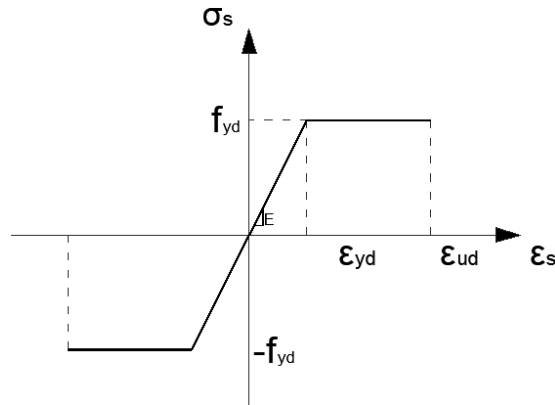


Figura 3.2: Diagrama tensão-extensão do aço

$$E_t(\varepsilon_s) = \begin{cases} 0 & , |\varepsilon_s| > \frac{f_{yd}}{E_s} \\ \frac{f_{yd}}{\varepsilon_{yd}} & , \text{caso contrário} \end{cases} \quad (3.4)$$

3.2 Equações governativas dos pontos elementares

O comportamento da secção é modelado discretizando a secção num conjunto de pequenas áreas, as quais são designadas por pontos materiais elementares ou, simplesmente, pontos materiais. Na presente secção são definidas as equações governativas dos pontos materiais necessárias à posterior aplicação dos métodos, que na sua integração geram as equações governativas da secção.

Tal como referido anteriormente, cada ponto material da secção recebe um conjunto de atributos fundamentais para a criação da secção transversal, que são:

- Material (betão ou aço);
- Área, A;
- Coordenadas, y e z, relativas ao centro de gravidade da secção. São também definidas variáveis locais que interagem com variáveis da secção, tais como deformações ou tensões.

3.2.1 Deformação

Admitindo a hipótese de Bernoulli, na qual as secções planas se mantêm planas, a extensão em cada ponto da secção pode ser calculada a partir de três deformações generalizadas definidas ao nível da secção, nomeadamente a extensão ε_G no centro de gravidade e as curvaturas χ_y e χ_z em torno dos eixos correspondentes. Em cada ponto material, a extensão correspondente é definida em (3.5).

$$\varepsilon = \varepsilon_G + \chi_y z - \chi_z y \quad (3.5)$$

Sejam \mathbf{a} e \mathbf{b} , os vetores definidos em (3.6).

$$\mathbf{a} = \begin{bmatrix} \varepsilon_G \\ \chi_y \\ \chi_z \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 1 \\ z \\ -y \end{bmatrix} \quad (3.6)$$

A extensão em cada ponto é definida pelo produto interno entre ambos os vetores, como indica a Equação (3.7). Os seus diferenciais parciais estão representados na Equação (3.8).

$$\varepsilon = \mathbf{a} \cdot \mathbf{b} \quad (3.7)$$

$$\frac{d\varepsilon}{d\varepsilon_G} = 1 \quad \frac{d\varepsilon}{d\chi_y} = z \quad \frac{d\varepsilon}{d\chi_z} = -y \quad (3.8)$$

Deste modo, a forma linearizada da extensão elementar em cada ponto é representada em (3.9)

$$d\varepsilon = d\varepsilon_G + d\chi_y z - d\chi_z y \quad (3.9)$$

ou (3.10)

$$d\varepsilon = d\mathbf{a} \cdot \mathbf{b} \quad (3.10)$$

3.2.2 Tensão

Para cada ponto material, a relação $\sigma(\varepsilon)$ depende do material e se o cálculo é não linear ou elástico. A linearização da Equação (3.11) é dada por (3.12).

$$\sigma = \sigma(\varepsilon) \quad (3.11)$$

$$d\sigma = E_t d\varepsilon \quad (3.12)$$

3.2.3 Esforços

Integrando as extensões elementares no domínio da secção, considerando a sua dependência direta com as deformações, são apresentados em (3.13) os esforços correspondentes.

$$N = \int_A \sigma dA \quad M_y = \int_A z \sigma dA \quad M_z = - \int_A y \sigma dA \quad (3.13)$$

Seja \mathbf{r} , em (3.14), o vetor dos esforços resistentes.

$$\mathbf{r} = \begin{bmatrix} N \\ M_y \\ M_z \end{bmatrix} \quad (3.14)$$

Linearizando (3.13), obtêm-se os esforços apresentados na equação (3.16). A título de exemplo, em (3.15), é deduzida a respetiva linearização para o esforço axial.

$$\begin{aligned} dN &= \int_A d\sigma dA = \int_A E_t d\varepsilon dA = \int_A E_t (d\varepsilon_G + d\chi_y z - d\chi_z y) dA \\ &= \left(\int_A E_t dA \right) d\varepsilon_G + \left(\int_A z E_t dA \right) d\chi_y + \left(\int_A -y E_t dA \right) d\chi_z \end{aligned} \quad (3.15)$$

$$\begin{bmatrix} dN \\ dM_y \\ dM_z \end{bmatrix} = \begin{bmatrix} \int_A E_t dA & \int_A z E_t dA & \int_A -y E_t dA \\ \int_A z E_t dA & \int_A z^2 E_t dA & \int_A -zy E_t dA \\ \int_A -y E_t dA & \int_A -zy E_t dA & \int_A y^2 E_t dA \end{bmatrix} \begin{bmatrix} d\varepsilon_G \\ d\chi_y \\ d\chi_z \end{bmatrix} \quad (3.16)$$

↓

$$\begin{bmatrix} dN \\ dM_y \\ dM_z \end{bmatrix} = \begin{bmatrix} K_{xx} & K_{xy} & K_{xz} \\ K_{yx} & K_{yy} & K_{yz} \\ K_{zx} & K_{zy} & K_{zz} \end{bmatrix} \begin{bmatrix} d\varepsilon_G \\ d\chi_y \\ d\chi_z \end{bmatrix}, \quad (3.17)$$

ou simplesmente, (3.18),

$$d\mathbf{r} = \mathbf{K} d\mathbf{a} \quad (3.18)$$

sendo \mathbf{K} a matriz de rigidez tangente da secção transversal.

3.2.4 Deformação máxima da secção transversal

Para determinar a resistência da secção para o estado limite último, devem ser respeitadas as deformações limites impostas pelos regulamentos, as quais são geralmente do tipo em (3.19).

$$\varepsilon_{min} < \varepsilon < \varepsilon_{max} \quad (3.19)$$

Nota para o facto da condição referida depender do material, atributo do ponto material. Desdobrando (3.19) com (3.5), a condição toma a forma definida em (3.20).

$$\varepsilon_{min} < \varepsilon_G + \chi_y z - \chi_z y < \varepsilon_{max} \quad \Rightarrow \quad \varepsilon_{min} < \mathbf{b} \cdot \mathbf{a} < \varepsilon_{max} \quad (3.20)$$

Nas secções de betão armado sujeitas apenas a compressões, o diagrama de deformações segue as regras esquematizadas na Figura 3.3 [1]. Com uma proporção de triângulos, em (3.21), é possível estabelecer as suas relações demonstradas em (3.22).

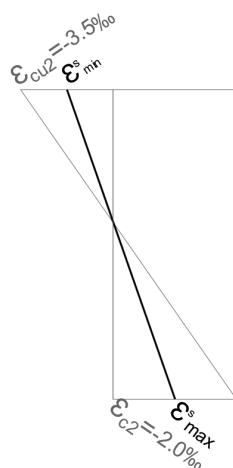


Figura 3.3: Diagrama de extensões para compressão pura

$$\frac{\varepsilon_{c2} - \varepsilon_{min}^S}{\varepsilon_{max}^S - \varepsilon_{c2}} = \frac{\varepsilon_{c2} - \varepsilon_{cu2}}{0 - \varepsilon_{c2}} \quad (3.21)$$

$$\begin{aligned} (\varepsilon_{c2} - \varepsilon_{min}^S)(0 - \varepsilon_{c2}) &= (\varepsilon_{c2} - \varepsilon_{cu2})(\varepsilon_{max}^S - \varepsilon_{c2}) \Leftrightarrow \\ \Leftrightarrow \varepsilon_{c2} \varepsilon_{min}^S - (\varepsilon_{c2} - \varepsilon_{cu2}) \varepsilon_{max}^S &= \varepsilon_{c2} \varepsilon_{cu2} \Leftrightarrow \\ \Leftrightarrow \varepsilon_{min}^S - (1 - \frac{\varepsilon_{cu2}}{\varepsilon_{c2}}) \varepsilon_{max}^S &= \varepsilon_{cu2} \end{aligned} \quad (3.22)$$

Atribuindo os valores máximos definidos em 3.3, obtém-se (3.23).

$$\begin{aligned} \varepsilon_{min}^S - (1 - \frac{-0.0035}{-0.0020})\varepsilon_{max}^S &= \varepsilon_{cu2} \Leftrightarrow \\ \Leftrightarrow \varepsilon_{min}^S + 0.75 \times \varepsilon_{max}^S &= \varepsilon_{cu2} \end{aligned} \quad (3.23)$$

Uma vez que ε_{min}^S e ε_{max}^S estão ambos à compressão, o valor máximo corresponderá a ε_{min}^S .

3.3 Geometria das secções transversais

Tal como foi explicado no início do capítulo, embora existam diferentes tipos de geometria com diversos graus de complexidade para as secções transversais, esta dissertação foca-se nas mais comuns. Contudo, qualquer combinação entre elas é possível. Independentemente do tipo de secção, todas recebem um conjunto de atributos, sejam eles a lista de pontos elementares já explicitados, as classes de betão e aço, ou o recobrimento da própria secção, entre outros. Resumidamente, o conjunto de pontos que a secção recebe, define geometricamente a mesma.

Por exemplo, a secção T é composta por duas secções retangulares, recorrendo a uma simples translação após determinação do centro gravítico do conjunto. Já no caso das secções circulares, estas serão divididas por um número determinado de coroas circulares, dependendo da malha, e cada coroa será dividida num número determinado de fatias.

3.3.1 Secção retangular

A secção retangular é definida por uma largura b e uma altura h , correspondendo à secção de betão, como ilustra a Figura 3.4.

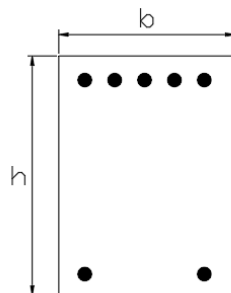


Figura 3.4: Secção retangular

Betão

A divisão da secção é efetuada em ambas as direções através de n_y e n_z , estes últimos ajustáveis de acordo com o refinamento desejado para a malha. Deste modo, cada ponto material terá uma área elementar definida em (3.24) e ilustrada em 3.5.

$$A_i = \frac{b}{n_y} \times \frac{h}{n_z} = d_y \times d_z \quad (3.24)$$

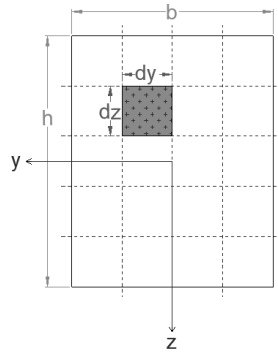


Figura 3.5: Subdivisão da secção em áreas elementares

O total de pontos materiais de betão é igual a $n_y \times n_z$ e cada ponto elementar recebe as suas coordenadas no centro gravítico da área elementar de betão, como ilustra 3.6, sendo esta uma aproximação uma vez que em caso algum a máxima extensão das fibras de betão se situa exatamente no centro gravítico do ponto, mas sim num dos vértices desse retângulo elementar. Porém, uma malha mais refinada resolve a aproximação e as diferenças serão praticamente inexistentes.

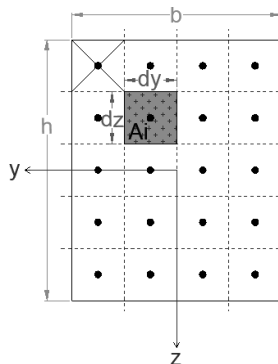


Figura 3.6: Centro gravítico dos retângulos elementares

Aço

Para que a secção transversal esteja completamente definida, é necessário colocar a armadura e definir o seu respetivo recobrimento. Os varões de aço podem ser colocados em qualquer uma das faces da secção, de forma independente, tal como ilustra a Figura 3.7.

Nesta configuração, cada varão de aço será tratado como um ponto elementar, com centro gravítico no centro do varão, e conseqüentemente a sua área é condensada no centróide do mesmo.

A secção retangular apresenta-se completamente definida.

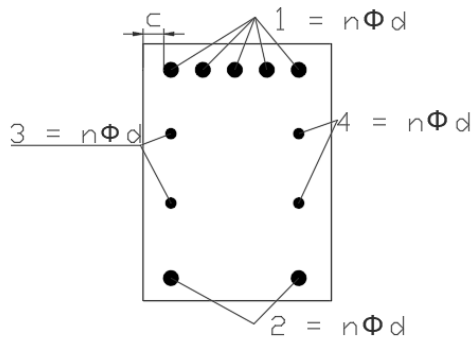


Figura 3.7: Posicionamento dos varões de aço

3.3.2 Secção T

Como ilustrado em 3.8, a secção T tem uma largura do banzo b_f e correspondente espessura t_f , uma altura da alma h_w e espessura t_w .

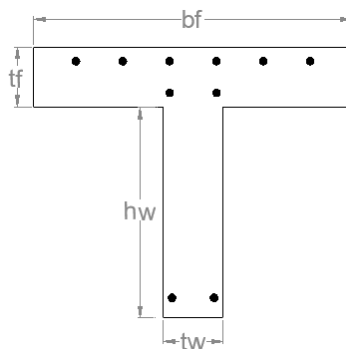


Figura 3.8: Secção T

Betão

A secção T, composta por dois retângulos, recebe todos os atributos de cada um deles, já explicitados na secção anterior, isto é, secção T = secção retangular R1 + secção retangular R2. Para R1, $b = b_f$ e $h = t_f$, e para R2, $b = t_w$ e $h = h_w$, como apresentado em 3.9.

Neste momento, ambos os retângulos têm o seu centro gravítico na origem do referencial. Após uma translação dos mesmos para o centróide do conjunto, a secção toma a forma definida em 3.10 e a secção, no que concerne ao material betão, estará completamente definida.

Aço

No posicionamento dos varões de aço, visto estes serem um atributo de cada um dos retângulos, é possível a sua colocação independente em cada uma das faces, ilustrado em 3.11. O centro gravítico da secção é reavaliado, e a secção T apresenta-se definida.

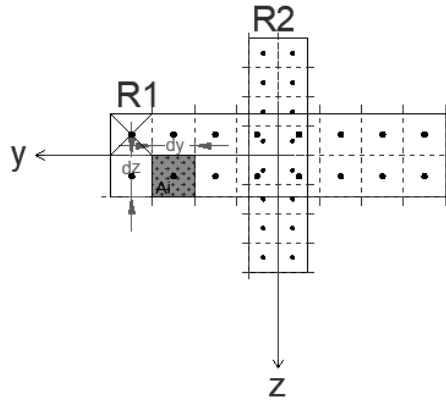


Figura 3.9: Secção $T = R1 + R2$

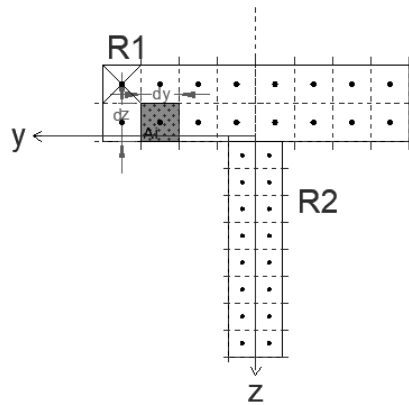


Figura 3.10: Centro gravítico da secção T

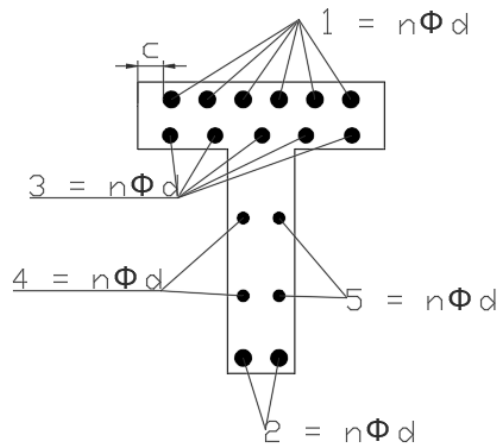


Figura 3.11: Posicionamento dos varões de aço

3.3.3 Secção circular

A secção circular define-se pelo seu diâmetro d , como ilustra 3.12.

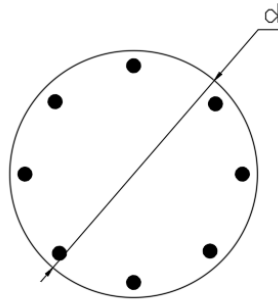


Figura 3.12: Secção circular

Betão

Contrariamente às secções anteriores, a secção circular, pela sua forma, será subdividida inicialmente em coroas circulares, com centro no centro da secção, e posteriormente subdividida em fatias, de forma a que cada pedaço elementar tenha sensivelmente a mesma área, como ilustra a Figura 3.13.

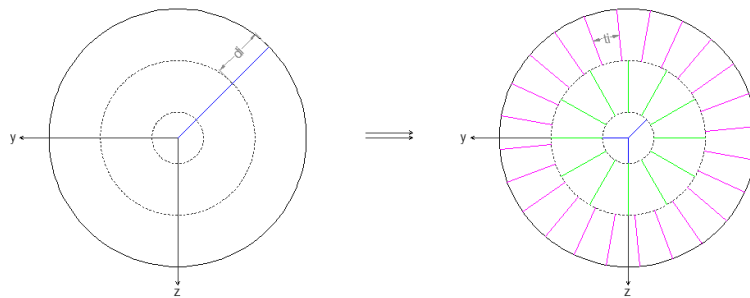


Figura 3.13: Subdivisão da secção circular

Esta será a principal diferença entre a secção circular e as acima mencionadas.

Aço

Para reforçar o betão, os varões de aço serão posicionados na periferia da secção circular com o respetivo recobrimento c , como indicado em 3.14, ficando a secção definida.

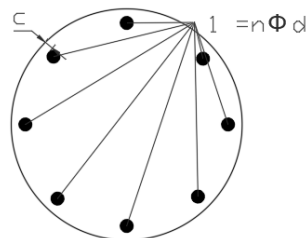


Figura 3.14: Posicionamento dos varões de aço

Capítulo 4

Superfícies resistentes de interação

O presente capítulo desenvolve vários métodos para a obtenção de um ponto da superfície de interação, métodos que serão utilizados no capítulo 5. A primeira subsecção uniformiza, de certo modo, a direção crescente de cada um dos métodos, baseando-a nos critérios de condição limite mencionados na secção 3.2.4 e extrapolados para toda a secção transversal.

Iniciando o percurso pelo primeiro método, que determina os esforços a partir das deformações dadas, passando pelo Método 2, que resumidamente dá um ponto da superfície de interação fixando um esforço axial N , e generalizando posteriormente para os Métodos 3 e 4 que, com um ponto inicial e uma direção, atingem a superfície de interação, o capítulo aborda todos os pontos-chave para a próxima etapa. É importante referir que os dois primeiros métodos servem quase exclusivamente para a formação dos seguintes, pelo que os maiores desenvolvimentos ocorrem nos métodos 3 e 4.

4.1 Deformações em estado limite último

Em muitas situações, e no caso dos métodos seguintes, as condições limite rege-se pela deformação máxima a aplicar à estrutura até que entre em rotura.

Admita-se que as deformações podem variar de uma linear como em (4.1) ou, condensadamente, em (4.2),

$$\begin{bmatrix} \varepsilon_G \\ \chi_y \\ \chi_z \end{bmatrix} = \begin{bmatrix} \varepsilon_G^{(0)} \\ \chi_y^{(0)} \\ \chi_z^{(0)} \end{bmatrix} + p \begin{bmatrix} \varepsilon_G^{(1)} \\ \chi_y^{(1)} \\ \chi_z^{(1)} \end{bmatrix} \quad (4.1)$$

$$\mathbf{a} = \mathbf{a}^{(0)} + p\mathbf{a}^{(1)} \quad (4.2)$$

onde as deformação referidas em (0) são as iniciais e as referidas em (1) indicam a direção da evolução. O valor do parâmetro p variará entre 0 e um valor p_{max} definido, por exemplo, pela condição limite (3.20).

Desdobrando as equações, obtém-se (4.3).

$$\begin{aligned} \varepsilon_{min} &< \mathbf{b} \cdot \mathbf{a} < \varepsilon_{max} \\ \varepsilon_{min} &< \mathbf{b} \cdot (\mathbf{a}^{(0)} + p\mathbf{a}^{(1)}) < \varepsilon_{max} \end{aligned} \quad (4.3)$$

Se $\mathbf{b} \cdot \mathbf{a}^{(1)} > 0$, isto é, a deformação no ponto aumenta com p , tem-se (4.4).

$$\frac{\varepsilon_{min} - \mathbf{b} \cdot \mathbf{a}^{(0)}}{\mathbf{b} \cdot \mathbf{a}^{(1)}} < p < \frac{\varepsilon_{max} - \mathbf{b} \cdot \mathbf{a}^{(0)}}{\mathbf{b} \cdot \mathbf{a}^{(1)}} \quad (4.4)$$

No caso de $\mathbf{b} \cdot \mathbf{a}^{(1)} < 0$, obtém-se (4.5).

$$\frac{\varepsilon_{max} - \mathbf{b} \cdot \mathbf{a}^{(0)}}{\mathbf{b} \cdot \mathbf{a}^{(1)}} < p < \frac{\varepsilon_{min} - \mathbf{b} \cdot \mathbf{a}^{(0)}}{\mathbf{b} \cdot \mathbf{a}^{(1)}} \quad (4.5)$$

Para o estado limite da zona 5 [1], mencionado em 3.2.4, uma vez que a secção é completamente sujeita a extensões negativas, é necessário identificar qual a extensão máxima, o que corresponde à deformação mínima na secção, dada por \mathbf{b} , em (4.6).

$$\mathbf{b}^{min} = \arg \min_b \{\mathbf{b} \cdot \mathbf{a}\} \quad (4.6)$$

Logo, as maior e menor deformações correspondem aos pontos indicados em (4.7).

$$\mathbf{b}^{min} = \arg \min_b \{\mathbf{b} \cdot \mathbf{a}\} \quad \mathbf{b}^{max} = \arg \max_b \{\mathbf{b} \cdot \mathbf{a}\} \quad (4.7)$$

A demonstração de \mathbf{b}^{z5} , usando a condição (3.23), é dada na equação (4.8), e tem a sua forma final em (4.9).

$$\begin{aligned} \varepsilon_{min}^S + 0.75\varepsilon_{max}^S &> \varepsilon_{cu2} \\ (\mathbf{b}^{min} \cdot (\mathbf{a}^{(0)} + p\mathbf{a}^{(1)})) + 0.75(\mathbf{b}^{max} \cdot (\mathbf{a}^{(0)} + p\mathbf{a}^{(1)})) &> \varepsilon_{cu2} \\ (\mathbf{b}^{min} + 0.75\mathbf{b}^{max}) \cdot \mathbf{a}^{(1)}p &> \varepsilon_{cu2} - (\mathbf{b}^{min} + 0.75\mathbf{b}^{max}) \cdot \mathbf{a}^{(0)} \\ \mathbf{b}^{z5} \cdot \mathbf{a}^{(1)}p &> \varepsilon_{cu2} - \mathbf{b}^{z5} \cdot \mathbf{a}^{(0)}, \end{aligned} \quad (4.8)$$

onde $\mathbf{b}^{z5} = \mathbf{b}^{min} + 0.75\mathbf{b}^{max}$.

$$\begin{aligned}
p &> \frac{\varepsilon_{cu2} - \mathbf{b}^{z5} \cdot \mathbf{a}^{(0)}}{\mathbf{b}^{z5} \cdot \mathbf{a}^{(1)}}, & \text{se } \mathbf{b}^{z5} \cdot \mathbf{a}^{(1)} > 0 \\
p &< \frac{\varepsilon_{cu2} - \mathbf{b}^{z5} \cdot \mathbf{a}^{(0)}}{\mathbf{b}^{z5} \cdot \mathbf{a}^{(1)}}, & \text{se } \mathbf{b}^{z5} \cdot \mathbf{a}^{(1)} < 0
\end{aligned} \tag{4.9}$$

4.2 Método 1 - Aplicando deformações

No caso das deformações serem conhecidas, é suficiente aplicar as fórmulas referidas em (3.9), (3.12) e (3.13).

Com o principal objetivo de desenhar a trajetória dos esforços correspondentes à aplicação de deformações crescentes, até atingir as condições limite (3.19) ou (3.23), este método simboliza o primeiro passo para uma avaliação generalizada das superfícies de interação resistentes.

A trajetória é representada na Equação (4.10),

$$\begin{bmatrix} \varepsilon_G \\ \chi_y \\ \chi_z \end{bmatrix} = \begin{bmatrix} \varepsilon_G^{(0)} \\ \chi_y^{(0)} \\ \chi_z^{(0)} \end{bmatrix} + p \begin{bmatrix} \varepsilon_G^{(1)} \\ \chi_y^{(1)} \\ \chi_z^{(1)} \end{bmatrix} \tag{4.10}$$

onde as deformações referidas como (0) são as iniciais e as referidas como (1) indicam a direção da trajetória. O valor de p variará entre 0 e um valor p_{max} determinado pelas condições limite.

4.3 Método 2 - Fixando N e θ

Quando a abordagem passa por se fixar $N = \bar{N}$ e a orientação da linha neutra $\theta = \bar{\theta}$, deixa de ser possível aplicar as equações diretamente, sendo necessário um processo iterativo para a determinação das deformações correspondentes ao estado limite último.

Sejam χ_y e χ_z , as curvaturas dadas em (4.11),

$$\chi_y = \chi \cos \theta \qquad \chi_z = \chi \sin \theta \tag{4.11}$$

onde θ é fixo e χ desconhecido.

A partir de um valor inicial de ε_G , e usando a condição limite, o valor máximo de χ obtém-se a partir de (4.12).

$$\varepsilon_{min} < \varepsilon = \varepsilon_G + \chi \cos \theta z - \chi \sin \theta y < \varepsilon_{max} \quad \Rightarrow \quad \frac{\varepsilon_{min} - \varepsilon_G}{\cos \theta z - \sin \theta y} < \chi < \frac{\varepsilon_{max} - \varepsilon_G}{\cos \theta z - \sin \theta y} \tag{4.12}$$

Aplicando a formulação da secção 4.1, obtém-se os parâmetros definidos em (4.13).

$$\mathbf{a}^{(0)} = \begin{bmatrix} \varepsilon_G \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{a}^{(1)} = \begin{bmatrix} 0 \\ \cos \theta \\ \sin \theta \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 1 \\ z \\ -y \end{bmatrix} \quad p = \chi \quad (4.13)$$

Desta forma, as curvaturas são funções de ε_G . Com os valores da deformação é possível calcular $N(\varepsilon_G, \chi_y(\varepsilon_G), \chi_z(\varepsilon_G))$. A função é definida em (4.14),

$$f(\varepsilon_G) = (\varepsilon_G, \chi_y(\varepsilon_G), \chi_z(\varepsilon_G)) - \bar{N} = 0 \quad (4.14)$$

cuja raiz pode ser obtida através de um método iterativo (método de Newton [4] ou método de *regula falsi* [10]).

Este método também permite o traçado de trajetórias, admitindo por exemplo χ variável e com um valor que esteja contido na superfície resistente da secção.

Note-se que é esta a metodologia habitualmente recomendada para a construção de diagramas de interação resistentes.

4.4 Método 3 - Definindo a direção dos esforços

Admitindo agora que os esforços aplicados são definidos por uma posição inicial (0) e uma direção dos mesmos (1), pode considerar-se uma função linear do tipo (4.15).

$$N(\lambda) = N^0 + \lambda N^{(1)} \quad M_y(\lambda) = M_y^0 + \lambda M_y^{(1)} \quad M_z(\lambda) = M_z^0 + \lambda M_z^{(1)} \quad (4.15)$$

Agrupando os esforços atuantes no vetor das solicitações s , em (4.16),

$$\mathbf{s} = \begin{bmatrix} N(\lambda) \\ M_y(\lambda) \\ M_z(\lambda) \end{bmatrix} \quad (4.16)$$

é possível compactar a equação, para o caso da função linear, em (4.17).

$$\mathbf{s} = \mathbf{s}^{(0)} + \lambda \mathbf{s}^{(1)} \quad (4.17)$$

Para resolver a equação, é necessário determinar o valor de λ no estado limite último e as correspondentes deformações.

Comece-se por considerar uma estimativa inicial, por exemplo $\lambda = 1$, $\varepsilon_G = 0$, $\chi_y = 0$ e $\chi_z = 0$. Seja \mathbf{a}^A o ponto definido pela estimativa. O vetor resíduo [12] terá a forma apresentada em (4.18) ou,

definido o vetor $\mathbf{r} = \begin{bmatrix} N(\varepsilon_G, \chi_y, \chi_z) \\ M_y(\varepsilon_G, \chi_y, \chi_z) \\ M_z(\varepsilon_G, \chi_y, \chi_z) \end{bmatrix}$, a forma compactada em (4.19).

$$\mathbf{R}^A = \begin{bmatrix} N(\varepsilon_G, \chi_y, \chi_z) - N(\lambda) \\ M_y(\varepsilon_G, \chi_y, \chi_z) - M_y(\lambda) \\ M_z(\varepsilon_G, \chi_y, \chi_z) - M_z(\lambda) \end{bmatrix} \quad (4.18)$$

$$\mathbf{R} = \mathbf{r} - \mathbf{s} \quad (4.19)$$

É necessário encontrar a solução \mathbf{a} que satisfaça $\mathbf{R}(\mathbf{a}) = 0$. Para utilizar o método de Newton, seja (4.20) a forma linearizada.

$$\mathbf{R}^A + \frac{d\mathbf{R}}{d\mathbf{a}}\Delta\mathbf{a} + \frac{d\mathbf{R}}{d\lambda}\Delta\lambda\dots = \mathbf{0} \quad (4.20)$$

As sensibilidades são dadas pela matriz $\frac{d\mathbf{R}}{d\mathbf{a}}$, que depende do ponto \mathbf{A} onde é aplicada, e pelo vetor fixo $\frac{d\mathbf{R}}{d\lambda}$, apresentados em (4.21),

$$\left[\frac{d\mathbf{R}}{d\mathbf{a}} \right] = \begin{bmatrix} K_{xx} & K_{xy} & K_{xz} \\ K_{yx} & K_{yy} & K_{yz} \\ K_{zx} & K_{zy} & K_{zz} \end{bmatrix} \quad \left[\frac{d\mathbf{R}}{d\lambda} \right] = \begin{bmatrix} -N^{(1)} \\ -M_y^{(1)} \\ -M_z^{(1)} \end{bmatrix} = -\mathbf{Q} \quad (4.21)$$

pelo que a solução é dada por (4.22),

$$\Delta\mathbf{a} = - \left[\frac{d\mathbf{R}}{d\mathbf{a}} \right]^{-1} (\mathbf{R}^A - \Delta\lambda\mathbf{Q}) = \mathbf{a}^R + \Delta\lambda\mathbf{a}^Q \quad (4.22)$$

onde \mathbf{a}^R , em (4.23), corresponde à correção a aplicar para λ constante e \mathbf{a}^Q , em (4.24), corresponde à correção a aplicar por se variar λ , mantendo o erro das equações.

$$\mathbf{a}^R = - \left[\frac{d\mathbf{R}}{d\mathbf{a}} \right]^{-1} \mathbf{R}^A = \begin{bmatrix} \varepsilon_G^R \\ \chi_y^R \\ \chi_z^R \end{bmatrix} \quad (4.23)$$

$$\mathbf{a}^Q = \left[\frac{d\mathbf{R}}{d\mathbf{a}} \right]^{-1} \mathbf{Q} = \begin{bmatrix} \varepsilon_G^Q \\ \chi_y^Q \\ \chi_z^Q \end{bmatrix} \quad (4.24)$$

As novas estimativas de \mathbf{a} e λ são apresentadas em (4.25).

$$\mathbf{a}^B = \mathbf{a}^A + \mathbf{a}^R + \Delta\lambda \mathbf{a}^Q \quad \lambda^B = \lambda^A + \Delta\lambda \quad (4.25)$$

O valor de $\Delta\lambda$ é calculado pelas condições limite (3.19). Para cada ponto da secção, a condição (4.26) deve ser respeitada.

$$\varepsilon_{min} < \mathbf{b} \cdot \mathbf{a} < \varepsilon_{max} \quad (4.26)$$

Substituindo o valor da nova estimativa, tem-se (4.27).

$$\varepsilon_{min} < \mathbf{b} \cdot (\mathbf{a}^A + \mathbf{a}^R + \Delta\lambda \mathbf{a}^Q) < \varepsilon_{max} \quad (4.27)$$

Se $\mathbf{b} \cdot \mathbf{a}^Q > 0$, o que significa que a deformação aumenta com λ , tem-se (4.28).

$$\frac{\varepsilon_{min} - \mathbf{b} \cdot (\mathbf{a}^A + \mathbf{a}^R)}{\mathbf{b} \cdot \mathbf{a}^Q} < \Delta\lambda < \frac{\varepsilon_{max} - \mathbf{b} \cdot (\mathbf{a}^A + \mathbf{a}^R)}{\mathbf{b} \cdot \mathbf{a}^Q} \quad (4.28)$$

Caso contrário, se $\mathbf{b} \cdot \mathbf{a}^Q < 0$, tem-se (4.29).

$$\frac{\varepsilon_{max} - \mathbf{b} \cdot (\mathbf{a}^A + \mathbf{a}^R)}{\mathbf{b} \cdot \mathbf{a}^Q} < \Delta\lambda < \frac{\varepsilon_{min} - \mathbf{b} \cdot (\mathbf{a}^A + \mathbf{a}^R)}{\mathbf{b} \cdot \mathbf{a}^Q} \quad (4.29)$$

Aplicando a metodologia da subsecção 4.1, apresentam-se, em (4.30), os parâmetros definidos.

$$\mathbf{a}^{(0)} = \mathbf{a}^A + \mathbf{a}^R \quad \mathbf{a}^{(1)} = \mathbf{a}^Q \quad p = \Delta\lambda \quad (4.30)$$

O $\Delta\lambda$ a adotar será o mínimo dos valores máximos obtidos para cada ponto material. Tendo obtido $\Delta\lambda$, fica definido o seguinte ponto a considerar no processo iterativo, pelas equações (4.25).

Note-se que esta técnica, baseada no cálculo das duas parcelas \mathbf{a}^R e \mathbf{a}^Q , é semelhante à utilizada nos métodos de controle (comprimentos de arco) usados para a determinação de trajetórias não lineares [12].

4.5 Método 4 - Variando a taxa de armadura

No caso das secções de betão armado, os esforços resistentes resultam da contribuição do betão e do aço nas armaduras. Como foi referido anteriormente, a área de cada varão de aço está concentrada no seu centro gravítico, pelo que as integrações do material são, simplifcadamente, proporcionais à taxa de armadura. Sendo assim, os esforços resistentes definem-se em (4.31).

$$\begin{aligned} N(\varepsilon_G, \chi_y, \chi_z, f_\omega) &= \int_A \sigma dA = N^b + f_\omega N^{a1} \\ M_y(\varepsilon_G, \chi_y, \chi_z, f_\omega) &= \int_A z \sigma dA = M_y^b + f_\omega M_y^{a1} \\ M_z(\varepsilon_G, \chi_y, \chi_z, f_\omega) &= - \int_A y \sigma dA = M_z^b + f_\omega M_z^{a1} \end{aligned} \quad (4.31)$$

As primeiras parcelas (b) correspondem à contribuição do betão e as segundas (a1) à contribuição do aço. O fator f_ω é multiplicado pelas armaduras iniciais, definidas pelo utilizador. As parcelas dos esforços são calculadas em (4.32).

$$\begin{aligned} N^b &= \int_{A_b} \sigma dA & M_y^b &= \int_{A_b} z \sigma dA & M_z^b &= - \int_{A_b} y \sigma dA \\ N^{a1} &= \int_{A_a} \sigma dA & M_y^{a1} &= \int_{A_a} z \sigma dA & M_z^{a1} &= - \int_{A_a} y \sigma dA \end{aligned} \quad (4.32)$$

Para além da dependência das deformações (ε_G , χ_y e χ_z), as parcelas dependem agora da taxa de armadura, ω . Deste modo, o vetor resíduo é definido em (4.33) e a sua forma linearizada apresentada em (4.34),

$$\mathbf{R}^a = \begin{bmatrix} N(\varepsilon_G, \chi_y, \chi_z, f_\omega) - \bar{N} \\ M_y(\varepsilon_G, \chi_y, \chi_z, f_\omega) - \bar{M}_y \\ M_z(\varepsilon_G, \chi_y, \chi_z, f_\omega) - \bar{M}_z \end{bmatrix} \quad (4.33)$$

$$\mathbf{R}^A + \frac{d\mathbf{R}}{da} \Delta a + \frac{d\mathbf{R}}{df_\omega} \Delta f_\omega \dots = \mathbf{0} \quad (4.34)$$

em que $\frac{d\mathbf{R}}{da} = K_{ij}$, desta vez calculada para o valor atual de f_ω , e $\frac{d\mathbf{R}}{df_\omega}$, apresentada em (4.35).

$$\left[\frac{d\mathbf{R}}{df_\omega} \right] = \begin{bmatrix} N^{a1} \\ M_y^{a1} \\ M_z^{a1} \end{bmatrix} = -\mathbf{Q} \quad (4.35)$$

O processo iterativo será, na sua maioria, idêntico ao da secção 4.4, substituindo $\Delta\lambda$ por Δf_ω , pelo que a solução é dada por (4.36),

$$\Delta \mathbf{a} = - \left[\frac{d\mathbf{R}}{d\mathbf{a}} \right]^{-1} (\mathbf{R}^A - \Delta f_\omega \mathbf{Q}) = \mathbf{a}^R + \Delta f_\omega \mathbf{a}^Q \quad (4.36)$$

onde \mathbf{a}^R , em (4.37), corresponde à correção a aplicar para f_ω constante e \mathbf{a}^Q , em (4.38), corresponde à correção a aplicar por se variar f_ω , mantendo o erro das equações.

$$\mathbf{a}_R = - \left[\frac{d\mathbf{R}}{d\mathbf{a}} \right]^{-1} \mathbf{R}^A = \begin{bmatrix} \varepsilon_G^R \\ \chi_y^R \\ \chi_z^R \end{bmatrix} \quad (4.37)$$

$$\mathbf{a}_Q = \left[\frac{d\mathbf{R}}{d\mathbf{a}} \right]^{-1} \mathbf{Q} = \begin{bmatrix} -\varepsilon_G^Q \\ -\chi_y^Q \\ -\chi_z^Q \end{bmatrix} \quad (4.38)$$

As novas estimativas de \mathbf{a} e f_ω são apresentadas em (4.39).

$$\mathbf{a}^B = \mathbf{a}^A + \mathbf{a}^R + \Delta f_\omega \mathbf{a}^Q \quad \lambda^B = \lambda^A + \Delta f_\omega \quad (4.39)$$

O valor de Δf_ω é calculado pelas condições limite (3.19). Para cada ponto da secção, a condição (4.40) deve ser respeitada.

$$\varepsilon_{min} < \mathbf{b} \cdot \mathbf{a} < \varepsilon_{max} \quad (4.40)$$

Substituindo o valor da nova estimativa, tem-se (4.41).

$$\varepsilon_{min} < \mathbf{b} \cdot (\mathbf{a}^A + \mathbf{a}^R + \Delta f_\omega \mathbf{a}^Q) < \varepsilon_{max} \quad (4.41)$$

Se $\mathbf{b} \cdot \mathbf{a}^Q > 0$, o que significa que a deformação aumenta com f_ω , tem-se (4.42).

$$\frac{\varepsilon_{min} - \mathbf{b} \cdot (\mathbf{a}^A + \mathbf{a}^R)}{\mathbf{b} \cdot \mathbf{a}^Q} < \Delta f_\omega < \frac{\varepsilon_{max} - \mathbf{b} \cdot (\mathbf{a}^A + \mathbf{a}^R)}{\mathbf{b} \cdot \mathbf{a}^Q} \quad (4.42)$$

Caso contrário, se $\mathbf{b} \cdot \mathbf{a}^Q < 0$, tem-se (4.43).

$$\frac{\varepsilon_{max} - \mathbf{b} \cdot (\mathbf{a}^A + \mathbf{a}^R)}{\mathbf{b} \cdot \mathbf{a}^Q} < \Delta f_\omega < \frac{\varepsilon_{min} - \mathbf{b} \cdot (\mathbf{a}^A + \mathbf{a}^R)}{\mathbf{b} \cdot \mathbf{a}^Q} \quad (4.43)$$

Aplicando a metodologia da subsecção 4.1, apresentam-se, em (4.44), os parâmetros definidos.

$$\mathbf{a}^{(0)} = \mathbf{a}^A + \mathbf{a}^R \quad \mathbf{a}^{(1)} = \mathbf{a}^Q \quad p = \Delta f_\omega \quad (4.44)$$

O Δf_ω a adotar será o mínimo dos valores máximos obtidos para cada ponto material. Tendo obtido Δf_ω , fica definido o seguinte ponto a considerar no processo iterativo, pelas equações (4.39).

Após concluído o processo, a taxa de armadura $\omega = \omega_0 \times f_\omega$ será a mínima necessária para resistir aos esforços atuantes.

4.5.1 Escolha da matriz de rigidez

A utilização da matriz de rigidez tangente nos métodos 3 e 4 pode não assegurar a convergência do processo iterativo. Por essa razão, é conveniente adotar regras heurísticas, baseadas na substituição da matriz de rigidez tangente por uma versão mais estável. Assim, quando o determinante da matriz de rigidez é muito baixo, sugere-se a utilização da matriz secante ou mesmo da matriz elástica. Estas matrizes obtêm-se como em (3.18), substituindo o valor do módulo tangente E_t pelo módulo secante $E_{sec} = \frac{\sigma}{\varepsilon}$ ou pelo módulo de elasticidade E .

4.6 Validação dos métodos

Os métodos são validados recorrendo a dois exemplos de flexão composta, um para a secção retangular e outro para a secção circular, e um último exemplo para a flexão composta desviada, com uma secção retangular. Em cada exemplo correlacionam-se os métodos entre si, e conseqüentemente são comparados com os ábacos de betão [5, 9, 11]. Estes ábacos foram construídos considerando um coeficiente $\alpha = 0.85$ para o valor de tensão de rotura de dimensionamento no betão. Deste modo, todos os exemplos se associarão ao mesmo coeficiente.

4.6.1 Exemplo 1 - Flexão composta - Secção retangular

A secção retangular considerada e as suas propriedades são definidas na Tabela 4.1.

Tabela 4.1: Secção transversal

Tipo	Betão	Aço	b [m]	h [m]	c [m]	R_{inf}	R_{sup}
Retangular	C30/37	A500	0.3	0.5	0.042	4 ϕ 16	4 ϕ 16

A discretização da secção terá $n_y = 20$ e $n_z = 20$, como ilustra a Figura 4.2, formada pelo script da Figura 4.1. Os esforços de dimensionamento são os definidos na Tabela 4.2.

Tabela 4.2: Esforços de dimensionamento

Ponto	N [kN]	M_y [kNm]
1	-2400	150
2	-2400	450
3	-4200	150
4	-1200	450
5	+1200	150

```
#Definição da secção transversal
C3037 = m.Betao('C30/37', alfa=0.85)
A500 = m.Aco('A500')

seccr = m.SeccaoR(0.3,0.5, C3037, A500, 20,20, arm_sup=[4,16], arm_inf=[4,16], c=0.042)
```

Figura 4.1: Script da secção transversal - Discretização

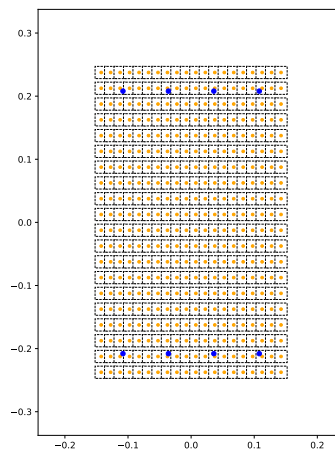


Figura 4.2: Secção transversal - Discretização

No presente exemplo recorre-se inicialmente aos ábacos de betão armado e posteriormente aos métodos e à sua correspondente validação, pela ordem inversa.

Ábacos de betão armado

Para definir o ábaco a utilizar no presente exemplo, $\frac{d_1}{h}$ é calculado em (4.45).

$$\frac{d_1}{h} = \frac{c + \frac{\phi}{2}}{h} = \frac{0.042 + \frac{0.016}{2}}{0.5} = 0.10 \quad (4.45)$$

Os esforços reduzidos, calculados pelas fórmulas (4.46), são determinados para 5 conjuntos de

esforços e apresentam-se na Tabela 4.3, bem como as correspondentes taxa de armadura ω e área de armadura A_{tot} , esta última definida em (4.47).

$$\nu = \frac{N}{bh f_{cd}} \quad (4.46)$$

$$\mu = \frac{M_y}{bh^2 f_{cd}}$$

$$A_{tot} = \omega \times bh \frac{f_{cd}}{f_{yd}} \quad (4.47)$$

Tabela 4.3: Taxa de armadura ω pelos ábacos de betão

Ponto	N [kN]	M_y [kNm]	ν	μ	ω	A_{tot} [cm ²]
1	-2400	150	-0.8	0.1	0.22	15.18
2	-2400	450	-0.8	0.3	0.78	53.82
3	-4200	150	-1.4	0.1	0.80	55.20
4	-1200	450	-0.4	0.3	0.50	30.36
5	+1200	150	+0.4	0.1	0.63	43.47

Métodos

O método 4 recebe de entrada os esforços definidos acima e obtém os resultados apresentados na Tabela 4.4 para a norma do vetor resíduo na ordem de 1×10^{-9} , através do *script* apresentado na Figura 4.3.

```
print('Method 4')
fw = secr.find_w(N=-2400, My=150, Mz=0, verbose=True)
fw = secr.find_w(N=-2400, My=450, Mz=0, verbose=True)
fw = secr.find_w(N=-4200, My=150, Mz=0, verbose=True)
fw = secr.find_w(N=-1200, My=450, Mz=0, verbose=True)
fw = secr.find_w(N=1200, My=150, Mz=0, verbose=True)
```

Figura 4.3: *Script* do método 4

Tabela 4.4: Resultados para o método 4

Ponto	$\varepsilon_G (\times 10^{-3})$	$\chi_y (\times 10^{-3})$	f_ω	ω	A_{tot} [cm ²]	$\frac{\omega_{\text{abaco}} - \omega_{\text{Método 4}}}{\omega_{\text{abaco}}} [\%]$	$n_{\text{iterações}}$
1	-1.77	7.30	0.918	0.214	14.76	2.7	15
2	-1.28	9.34	3.249	0.757	52.26	2.9	29
3	-1.87	3.80	3.419	0.797	54.99	0.4	13
4	-0.48	12.72	2.030	0.473	32.65	5.4	11
5	+5.29	22.66	2.745	0.640	44.15	1.6	226

O número de iterações para o ponto 5 é cerca de 20 vezes superior aos restantes. A convergência para a raiz do vetor resíduo é mais lenta uma vez que, para um esforço axial de tração, a tensão do

betão é considerada nula. Os métodos heurísticos utilizados na escolha da matriz de rigidez não foram calibrados para esta situação.

O método 3 representa o fator de segurança da secção relativamente aos esforços de dimensionamento. Nesta linha de pensamento, o fator λ deverá ser aproximadamente 1 se a área de armadura corresponder às anteriormente calculadas pelo método 4 na Tabela 4.4, isto é, se a área de armadura da secção inicial for multiplicada pelo fator f_w . A tabela 4.5 mostra os seus resultados com uma tolerância de erro de 1×10^{-9} , através do *script* representado na Figura 4.4.

```
fator, it1 = secr.find_lambda_Advanced(-2400,150,0, fw=0.918)
fator, it2 = secr.find_lambda_Advanced(-2400,450,0, fw=3.249)
fator, it3 = secr.find_lambda_Advanced(-4200,150,0, fw=3.4188)
fator, it4 = secr.find_lambda_Advanced(-1200,450,0, fw=2.03)
fator, it5 = secr.find_lambda_Advanced(1200,150,0, fw=2.745)
```

Figura 4.4: *Script* do método 3

Tabela 4.5: Resultados para o método 3

Ponto	$\varepsilon_G (\times 10^{-3})$	$\chi_y (\times 10^{-3})$	λ	$n_{\text{iterações}}$
1	-1.75	7.24	1.000	20
2	-1.28	9.34	1.000	14
3	-1.87	3.80	1.000	31
4	-0.48	12.72	1.000	15
5	5.29	22.66	1.000	8

Prosseguindo para o método 2, fixando o valor do esforço axial $N = \bar{N}$ e $\theta = 0$, com uma área de armadura correspondente à determinada pelo método 4, obtêm-se os valores correspondentes de M_y na Tabela 4.6, através do *script* ilustrado em 4.5.

```
#Método 2
N, My, Mz=secr.metodo_2(-2400, 0)
N, My, Mz=secr.metodo_2(-2400, 0)
N, My, Mz=secr.metodo_2(-4200, 0)
N, My, Mz=secr.metodo_2(-1200, 0)
N, My, Mz=secr.metodo_2(1200, 0)
```

Figura 4.5: *Script* do método 2

Tabela 4.6: Resultados para o método 2

Ponto	$N = \bar{N} [kN]$	$M_y [kNm]$	$n_{\text{iterações}}$
1	-2400	150.00	12
2	-2400	450.04	17
3	-4200	150.05	13
4	-1200	449.97	10
5	+1200	150.03	12

A partir das deformações resultantes do método 4, apresentam-se os esforços atuantes calculados pelo método 1, na Tabela 4.7, com uma área de armadura igual à obtida pelo método 4.

Tabela 4.7: Resultados para o método 1

Ponto	$N = \bar{N} [kN]$	$M_y [kNm]$
1	-2400.05	149.99
2	-2397.42	450.40
3	-4198.67	150.30
4	-1199.68	450.03
5	+1202.98	149.39

A correlação entre métodos é evidente. A maior diferença reside na comparação entre a armadura no ábaco e a armadura calculada pelo método 4, com o erro médio interno calculado de 2.6%. Este erro deve-se especialmente à baixa precisão na definição da taxa de armadura retirada do ábaco. No entanto, é um valor aceitável. Espera-se, no segundo exemplo, obter um erro da mesma ordem de grandeza.

4.6.2 Exemplo 2 - Flexão composta - Secção circular

A secção circular considerada e as suas propriedades são definidas na Tabela 4.8.

Tabela 4.8: Secção transversal

Tipo	Betão	Aço	d [m]	c [m]	R_{tot}
Circular	C30/37	A500	0.8	0.074	20 ϕ 12

A discretização da secção, programa em 4.6 e ilustrada em 4.7, tem 8 coroas circulares e para cada uma, o número de fatias é $n_i \times 4$, com $n_i \in \{1 : 8\}$, sendo n_0 a coroa menor. Os esforços de dimensionamento para os 5 conjuntos são os definidos na Tabela 4.9.

```
#Definição da secção transversal
C3037 = m.Betao('C30/37', alfa=0.85)
A500 = m.Aco('A500')
secc = m.SeccaoC(0.8, C3037, A500, dr=0.05,
                armadura=[20,16], c=0.074)
```

Figura 4.6: Script da secção transversal - Discretização

Tabela 4.9: Esforços de dimensionamento

Ponto	N [kN]	M_y [kNm]
1	-8042.48	804.25
2	-8042.48	2412.74
3	-16084.96	804.25
4	+4021.24	804.25
5	+4021.24	2412.74

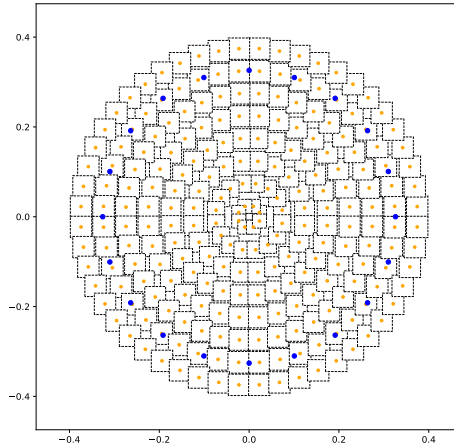


Figura 4.7: Secção transversal - Discretização

Ábacos de betão armado

Para definir o ábaco a utilizar no presente exemplo, $\frac{d_1}{h}$ é calculado em (4.48).

$$\frac{d_1}{h} = \frac{c + \frac{\phi}{2}}{h} = \frac{0.074 + \frac{0.012}{2}}{0.5} = 0.10 \quad (4.48)$$

Os esforços reduzidos, calculados pelas fórmulas (4.49), são determinados para 5 conjuntos de esforços e apresentam-se na Tabela 4.10, bem como as correspondentes taxa de armadura ω e área de armadura A_{tot} , esta última definida em (4.50).

$$\nu = \frac{N}{\pi\left(\frac{\phi}{2}\right)^2 f_{cd}} \quad (4.49)$$

$$\mu = \frac{M_y}{2\pi\left(\frac{\phi}{2}\right)^3 f_{cd}}$$

$$A_{tot} = \omega \times \pi\left(\frac{\phi}{2}\right)^2 \frac{f_{cd}}{f_{yd}} \quad (4.50)$$

Tabela 4.10: Taxa de armadura ω pelos ábacos de betão

Ponto	N [kN]	M_y [kNm]	ν	μ	ω	A_{tot} [cm ²]
1	-8042.48	804.25	-0.8	0.1	0.30	69.37
2	-2010.62	2412.74	-0.2	0.3	0.97	231.22
3	-16084.96	804.25	-1.6	0.1	1.10	254.34
4	+4021.24	804.25	+0.4	0.1	0.67	154.92
5	+2010.62	804.25	+0.2	0.1	0.48	110.99

Métodos

O método 4 recebe de entrada os esforços definidos acima e obtém os resultados apresentados na Tabela 4.11 para a norma do vetor resíduo na ordem de 1×10^{-9} , a partir do *script* ilustrado em 4.8.

```
print('Method 4')
fw = secc.find_w(N=-8042.48, My=804.25, Mz=0, verbose=True)
fw = secc.find_w(N=-2010.62, My=2412.74, Mz=0, verbose=True)
fw = secc.find_w(N=-16084.96, My=804.25, Mz=0, verbose=True)
fw = secc.find_w(N=4021.24, My=804.25, Mz=0, verbose=True)
fw = secc.find_w(N=2010.62, My=804.25, Mz=0, verbose=True)
```

Figura 4.8: *Script* do método 4

Tabela 4.11: Resultados para o método 4

Ponto	$\varepsilon_G (\times 10^{-3})$	$\chi_y (\times 10^{-3})$	f_ω	ω	$A_{tot} [\text{cm}^2]$	$\frac{\omega_{\acute{a}baco} - \omega_{M\acute{e}todo\ 4}}{\omega_{\acute{a}baco}} [\%]$	$n_{\text{iterações}}$
1	-1.51	5.32	3.043	0.298	68.82	0.7	12
2	+0.46	10.57	9.499	0.929	214.85	4.4	18
3	-1.82	3.31	6.261	1.090	251.76	1.0	8
4	+4.56	16.68	6.843	0.669	154.79	0.1	31
5	+4.29	17.53	4.816	0.471	108.92	1.9	11

O método 3 representa o fator de segurança da secção relativamente aos esforços de dimensionamento. Nesta linha de pensamento, o fator λ deverá ser aproximadamente 1 se a área de armadura corresponder às anteriormente calculadas pelo método 4 na Tabela 4.11, isto é, se a área de armadura da secção inicial for multiplicada pelo fator f_ω . A tabela 4.12 mostra os seus resultados com uma tolerância de erro de 1×10^{-9} , com o *script* em 4.9.

```
secc = m.SeccaoC(0.8, C3037, A500, dr=0.05, armadura=[20,29.6014], c=0.074)
secc = m.SeccaoC(0.8, C3037, A500, dr=0.05, armadura=[20,52.30253], c=0.074)
secc = m.SeccaoC(0.8, C3037, A500, dr=0.05, armadura=[20,40.03440944], c=0.074)
secc = m.SeccaoC(0.8, C3037, A500, dr=0.05, armadura=[20,44.39422813], c=0.074)
secc = m.SeccaoC(0.8, C3037, A500, dr=0.05, armadura=[20,37.23993169], c=0.074)
```

Figura 4.9: *Script* do método 3

Tabela 4.12: Resultados para o método 3

Ponto	$\varepsilon_G (\times 10^{-3})$	$\chi_y (\times 10^{-3})$	λ	$n_{\text{iterações}}$
1	-1.51	5.32	1.000	16
2	+0.46	10.57	1.000	37
3	-1.82	3.31	1.000	28
4	+4.56	16.68	1.000	19
5	+4.29	17.52	1.000	26

Prosseguindo para o método 2, fixando o valor do esforço axial $N = \bar{N}$ e $\theta = 0$, com uma área de armadura correspondente à determinada pelo método 4, obtêm-se os valores correspondentes de M_y na Tabela 4.13, pelo *Script* em 4.10.

```

#Método 2
N, My, Mz=secc.metodo_2(-8042.48, 0)
N, My, Mz=secc.metodo_2(-2010.62, 0)
N, My, Mz=secc.metodo_2(-16084.96, 0)
N, My, Mz=secc.metodo_2(4021.24, 0)
N, My, Mz=secc.metodo_2(2010.62, 0)

```

Figura 4.10: *Script* do método 2

Tabela 4.13: Resultados para o método 2

Ponto	$N = \bar{N}[kN]$	$M_y[kNm]$	$n_{\text{iterações}}$
1	-8042.48	804.24	15
2	-2010.62	2412.73	13
3	-16084.96	804.27	13
4	+4021.24	804.27	35
5	+2010.62	804.21	25

A partir das deformações resultantes do método 4, apresentam-se os esforços atuantes calculados pelo método 1, na Tabela 4.14, com uma área de armadura igual à obtida pelo método 4.

Tabela 4.14: Resultados para o método 1

Ponto	$N = \bar{N}[kN]$	$M_y[kNm]$
1	-8043.80	804.03
2	-2010.32	2412.76
3	-16087.11	803.81
4	+4021.64	804.16
5	+2008.59	804.77

A correlação entre métodos é evidente. A maior diferença reside na comparação entre a armadura no ábaco e a armadura calculada pelo método 4, com o erro médio calculado de 1.6%. Este erro ocorre especialmente devido à baixa precisão na definição da taxa de armadura retirada do ábaco e a erros de aproximação. No entanto permanece na mesma ordem de grandeza, tal como esperado.

4.6.3 Exemplo 3 - Flexão composta desviada - Secção retangular

O exemplo 3 determina somente um conjunto de esforços, definido em 4.16, uma vez que já se correlacionaram os métodos e foram verificados com os ábacos de betão armado. A secção transversal considerada é retangular e as suas propriedades são definidas na Tabela 4.15.

Tabela 4.15: Secção transversal

Tipo	Betão	Aço	b [m]	h [m]	c [m]	R_{inf}	R_{sup}
Retangular	C30/37	A500	0.5	0.5	0.042	4 ϕ 16	4 ϕ 16

A discretização da secção terá $n_y = 20$ e $n_z = 20$, como ilustra a Figura 4.12, através do *Script* em 4.11.

```

#Definição da secção transversal
C3037 = m.Betao('C30/37', alfa=0.85)
A500 = m.Aco('A500')

secc = m.SeccaoR(0.5,0.5, C3037, A500, 20,20, arm_sup=[4,16], arm_inf=[4,16], c=0.042)

```

Figura 4.11: Script da secção transversal - Discretização

Tabela 4.16: Esforços de dimensionamento

Ponto	N [kN]	M_y [kNm]	M_z [kNm]
1	-2000	750	250

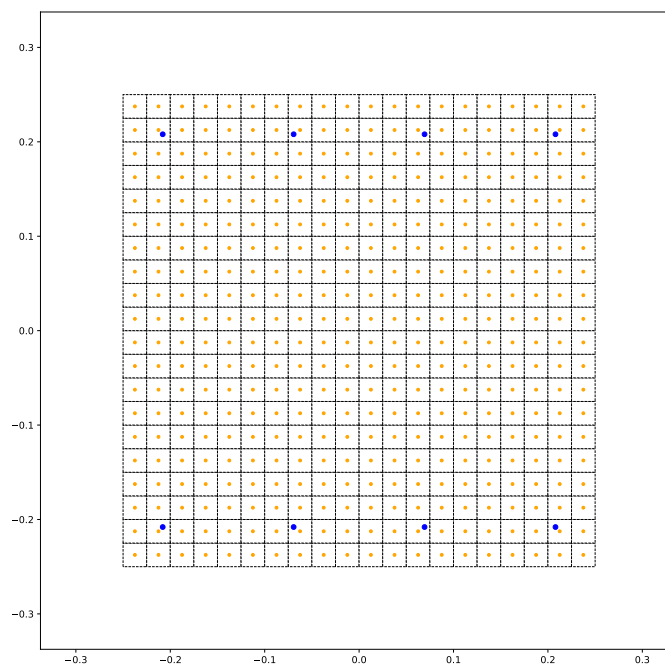


Figura 4.12: Secção transversal - Discretização

Ábacos de betão armado

Para definir o ábaco a utilizar no exemplo em causa, e uma vez que $b = h$, $\frac{b_1}{h} = \frac{d_1}{h}$ é calculado em (4.51).

$$\frac{b_1}{h} = \frac{d_1}{h} = \frac{c + \frac{\phi}{2}}{h} = \frac{0.042 + \frac{0.016}{2}}{0.5} = 0.10 \quad (4.51)$$

Os esforços reduzidos, calculados pelas fórmulas (4.52), são determinados para o conjunto de esforços e apresentam-se na Tabela 4.17, bem como as correspondentes taxa de armadura ω e área de armadura A_{tot} , esta última definida em (4.53). Tendo em consideração que $\mu_y > \mu_z$, então $\mu_1 = \mu_y$

e $\mu_2 = \mu_z$

$$\begin{aligned}\nu &= \frac{N}{bh f_{cd}} \\ \mu_y &= \frac{M_y}{bh^2 f_{cd}} \\ \mu_z &= \frac{M_z}{b^2 h f_{cd}}\end{aligned}\tag{4.52}$$

$$A_{tot} = \omega \times bh \frac{f_{cd}}{f_{yd}}\tag{4.53}$$

Tabela 4.17: Taxa de armadura ω pelos ábacos de betão

Ponto	N [kN]	M_y [kNm]	M_z [kNm]	ν	μ_y	μ_z	ω	A_{tot} [cm ²]
1	-2000	750	250	-0.4	0.3	0.1	0.70	80.50

Métodos

O método 4 recebe de entrada os esforços definidos acima e obtém os resultados apresentados na Tabela 4.18 pelo *Script* em 4.13 para a norma do vetor resíduo na ordem de 1×10^{-9} .

```
print('Method 4')
fw = secr.find_w(N=-2000, My=750, Mz=250, verbose=True)
```

Figura 4.13: *Script* do método 4

Tabela 4.18: Resultados para o método 4

Ponto	$\varepsilon_G (\times 10^{-3})$	$\chi_y (\times 10^{-3})$	$\chi_z (\times 10^{-3})$	f_ω	ω	A_{tot} [cm ²]	$\frac{\omega_{\acute{a}baco} - \omega_{\acute{a}baco\ M\acute{e}t\acute{o}d\acute{o}\ 4}}{\omega_{\acute{a}baco}}$ [%]	$n_{\text{iterações}}$
1	-0.26	8.49	5.15	5.113	0.715	82.24	2.143	8

O método 3 representa o fator de segurança da secção relativamente aos esforços de dimensionamento. Nesta linha de pensamento, o fator λ deverá ser aproximadamente 1 se a área de armadura corresponder às anteriormente calculadas pelo método 4 na Tabela 4.18, isto é, se a área de armadura da secção inicial for multiplicada pelo fator f_ω . A tabela 4.19 mostra os resultados do *script* em 4.14 para uma tolerância de 1×10^{-9} .

```
print('Method 3')
fator = secr.find_lambda_Advanced(-2000, 750, 250, verbose=True, fw=5.1131)
```

Figura 4.14: *Script* do método 3

Tabela 4.19: Resultados para o método 3

Ponto	$\varepsilon_G (\times 10^{-3})$	$\chi_y (\times 10^{-3})$	$\chi_z (\times 10^{-3})$	λ	$n_{\text{iterações}}$
1	-0.26	8.49	5.15	1.000	26

O método 2 não é aplicável a menos que seja previamente determinado o valor de θ pelo que não será avaliado neste exemplo, já que foi correlacionado anteriormente. Note que só não é aplicável porque o método 2 é um método base, sendo que os métodos 3 e 4 são os extrapolados para os capítulos seguintes, e por essa razão têm maior versatilidade.

A partir das deformações resultantes do método 4, apresentam-se os esforços atuantes calculados pelo método 1, na Tabela 4.20, com uma área de armadura igual à obtida pelo método 4.

Tabela 4.20: Resultados para o método 1

Ponto	$N = \bar{N}[kN]$	$M_y[kNm]$	$M_z[kNm]$	$n_{\text{iterações}}$
1	-2000.77	749.99	249.98	8

O erro associado entre o ábaco e o método 4 é de 2.2%, mesma ordem de grandeza dos exemplos anteriores. Não omitindo o erro associado, acredita-se plenamente que este esteja associado uma vez mais à falta de precisão na definição de ω a partir do ábaco e por erros computacionais de arredondamento. Sendo assim, dão-se por validados os métodos do capítulo 4.

Capítulo 5

Esforços resultantes de uma análise sísmica por espectros de resposta

Os métodos anteriores determinam, na sua generalidade, um ponto da superfície resistente, através de uma direção. É necessário entender qual a direção crítica da superfície resistente. As secções seguintes abordam os primeiros passos para a sua concretização.

5.1 Diagramas de resposta

Para o caso de os esforços atuantes resultarem de uma ação sísmica analisada por espectros de resposta, é necessário obter o seu diagrama de interação [13]. Sejam $s^{(0)}$ os esforços atuantes de uma análise estática para as cargas permanentes e sejam $s_k^{(i)}$ os esforços da resposta do modo i da estrutura devido à componente de base k segundo uma análise dinâmica modal. Admita-se que os vetores s têm três componentes, ou seja, o diagrama de interação será tridimensional, com $m = 3$.

A resposta dinâmica total poderá ser obtida aplicando regras de combinação quadráticas, como *CQC* para combinar modos e *SRSS* para combinar componentes de base. Para considerar a interação entre esforços, é conveniente aplicar estas últimas a combinações lineares do tipo de (5.1),

$$s_t = s \cdot t \quad (5.1)$$

sendo que o vetor t contém os fatores da combinação linear.

A resposta dinâmica na direção t é definida em (5.2),

$$t \cdot s \leq s_t^{(CQC+SRSS)} = \sqrt{\sum_{k=1}^{n_g} \sum_{i=1}^n \sum_{j=1}^n \rho_{ij} (t \cdot s_k^{(i)}) (t \cdot s_k^{(j)})} \quad (5.2)$$

onde n é o número de modos, n_g o número de componentes de base, ρ_{ij} os coeficientes de correlação modal definidas pela regra CQC e que dependem fundamentalmente da relação entre frequências de cada par de modos.

Evidenciando t , a equação pode ser escrita do tipo em (5.3).

$$\mathbf{s}_t^{(CQC+SRSS)} = \sqrt{\mathbf{t} \cdot \left(\sum_{k=1}^{n_g} \sum_{i=1}^n \sum_{j=1}^n \rho_{ij} (\mathbf{s}_k^{(i)} \otimes \mathbf{s}_k^{(j)}) \right) \mathbf{t}} \quad (5.3)$$

Definindo a matriz S , matriz de correlação $m \times m$ em (5.4),

$$S = \sum_{k=1}^{n_g} \sum_{i=1}^n \sum_{j=1}^n \rho_{ij} (\mathbf{s}_k^{(i)} \otimes \mathbf{s}_k^{(j)}) \quad (5.4)$$

a regra de combinação pode ser compactada para (5.5).

$$\mathbf{t} \cdot \mathbf{s} \leq s_t^{(CQC+SRSS)} = \sqrt{\mathbf{t} \cdot S \mathbf{t}} \quad (5.5)$$

Admita-se que o espaço de resposta não contém componentes linearmente independentes e que por isso a matriz S é invertível e $s_t^{(CQC+SRSS)} > 0$ para qualquer direção t .

O conjunto das condições (5.5) escritas para todas as direções t permite definir um diagrama de resposta elipsoidal. Pode mostrar-se que o ponto crítico desse elipsoide na direção t é dado por (5.6),

$$\mathbf{s}^{(CQC+SRSS)} = \mathbf{s}^{(0)} + \frac{S \mathbf{t}}{\sqrt{S \mathbf{t} \cdot \mathbf{t}}} \quad (5.6)$$

onde está incluída a contribuição estática. O elipsoide é, portanto, tangente a uma linha/plano perpendicular a t . Note-se que embora o comprimento do vetor t seja irrelevante, uma vez que é afetado no numerador e no denominador, o seu sentido é importante. É também possível obter o ponto de interseção do elipsoide com uma dada direção v através de (5.7). Na figura 5.1 ilustra-se a relação entre os vetores v e t e a parcela sísmica de s .

$$\mathbf{s}^{(CQC+SRSS)} = \mathbf{s}^{(0)} + \frac{\mathbf{v}}{\sqrt{\mathbf{v} \cdot S^{-1} \mathbf{v}}} \quad (5.7)$$

5.2 Extrapolação de métodos para ações sísmicas

Para extrapolar os métodos anteriores para as superfícies de interação, são duas as abordagens existentes, ou a verificação de segurança, ou o seu dimensionamento, otimizando deste modo a secção.

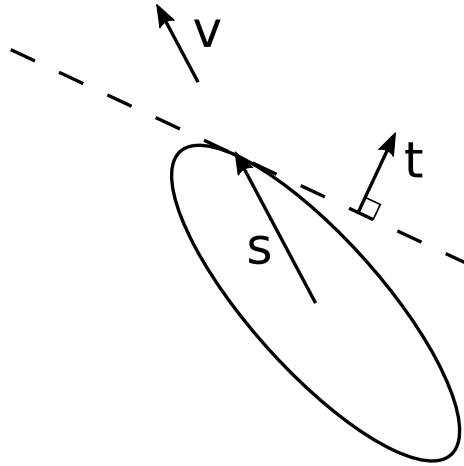


Figura 5.1: Superfície de interação da resposta sísmica - relação entre os vetores v e t

5.2.1 Método 5

Fazendo sentido considerar uma ação permanente fixa, mas a resposta sísmica afetada de um parâmetro λ , tem-se (5.8),

$$s^{(CQC+SRSS)} = s^{(0)} + \lambda \frac{St}{\sqrt{St \cdot t}} \quad (5.8)$$

completamente análoga a (4.17), sendo $s^{(1)} = \frac{St}{\sqrt{St \cdot t}}$, desde que o vetor t seja considerado fixo.

A verificação da segurança de uma secção envolve a aplicação do método 3 para um conjunto representativo de t direções, sendo portanto importante verificar se o menor dos valores λ é ainda superior a 1. No seu essencial, este método rege-se pela estratégia SED considerada por Sessa et al. [15], embora considerem o envelope supremo da regra CQC3 e utilizem uma estratégia secante para a obtenção de λ . Na prática, a superfície de interação será aumentada por um fator λ até que esta interseje a superfície resistente.

5.2.2 Método 6

Se o dimensionamento da secção é o objetivo principal do tema, tem-se (5.9),

$$s^{(CQC+SRSS)} = s^{(0)} + 1 \times \frac{St}{\sqrt{St \cdot t}} \quad (5.9)$$

isto é, o parâmetro λ recebe o valor fixo de 1. Neste caso a superfície de interação não se altera, ao contrário da superfície resistente. Dever-se-á, portanto, considerar o método 4 para cada t escolhendo o maior dos valores mínimos obtidos. Posteriormente, o método 3 será chamado para desenhar a superfície resistente com um $f_{\omega} = f_{\omega}^{min}$.

5.2.3 Validação de métodos

A secção transversal resistente definida para o exemplo é ilustrada em 5.2, com $n_y = n_z = 10$ e tem as propriedades descritas na Tabela 5.1.

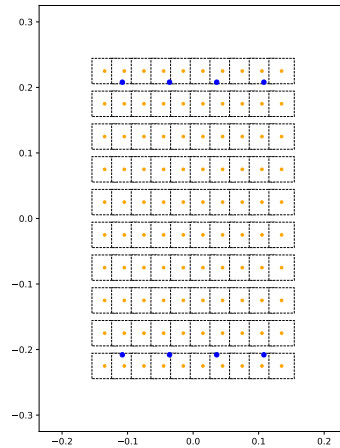


Figura 5.2: Secção transversal

Tabela 5.1: Secção transversal

Tipo	Betão	Aço	b [m]	h [m]	c [m]	R_{inf}	R_{sup}
Retangular	C30/37	A500	0.3	0.5	0.042	$4\phi 16$	$4\phi 16$

Seja, por hipótese, a superfície ilustrada em 5.3, a superfície de interação da ação sísmica, SI.

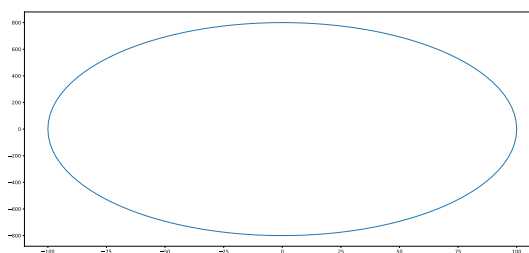


Figura 5.3: Superfície de interação da ação sísmica (SI)

Método 5

Sobrepondo a superfície de interação à superfície resistente inicial $SR_{inicial}$, para $\lambda = 1$, definida pela secção transversal, obtém-se a Figura 5.4. Como é evidente, a superfície de interação não está completamente contida no interior da superfície resistente, pelo que o fator λ terá de ser inferior a 1, o que significa que um sismo desta intensidade provocaria rotura na estrutura.

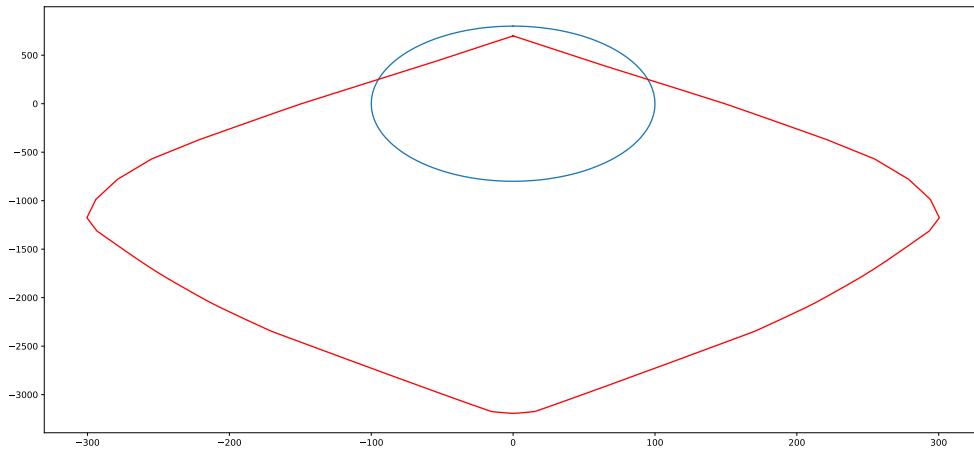


Figura 5.4: Sobreposição $SI-SR_{inicial}$

Com o *script* definido na Figura 5.5, o método determina o valor de $\lambda_{max} = 0.749$ e na Figura 5.6 a curva a verde é a superfície de interação máxima para que a secção resista. O parâmetro NsM é um fator de escala descrevendo a relação entre as ordens de grandeza de N e de M , cuja única função é obter pontos bem distribuídos no desenho.

```
##-----MAIN-----
C3037 = Betao('C30/37', alfa=0.85)
A500 = Aco('A500')

secr = SeccaoR(0.3,0.5, C3037, A500, 10,10, arm_sup=[4,16],
              arm_inf=[4,16], arm_latE=[0,0], arm_latD=[0,0],c=0.042)
secr.met5_lambda(NsM=.1)
```

Figura 5.5: *Script*

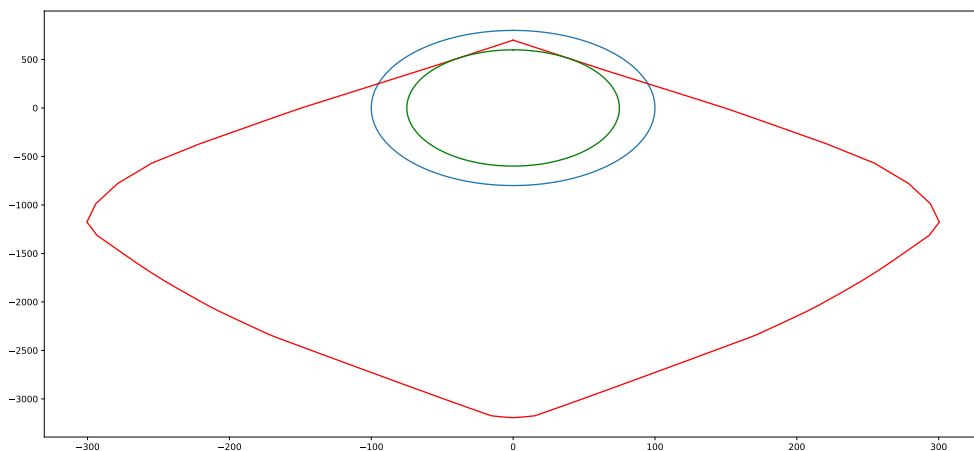


Figura 5.6: Sobreposição $SI - SR_{inicial} - SR_{ótima}$

Deste modo, a Equação (5.9) é, agora, (5.10).

$$s^{(CQC+SRSS)} = s^{(0)} + 0.749 \times \frac{St}{\sqrt{St \cdot t}} \quad (5.10)$$

Método 6

Para a mesma ação sísmica, e utilizando o método 6, é fácil concluir que a secção resistente terá de ser aumentada para que possa resistir à intensidade da ação sísmica, de acordo com os resultados obtidos pelo método 5. De facto, sobrepondo a superfície de interação à superfície resistente inicial $SR_{inicial}$ em 5.7, para $f_{\omega} = 1$, dever-se-á obter a mesma figura de 5.4. Como é evidente, o domínio da

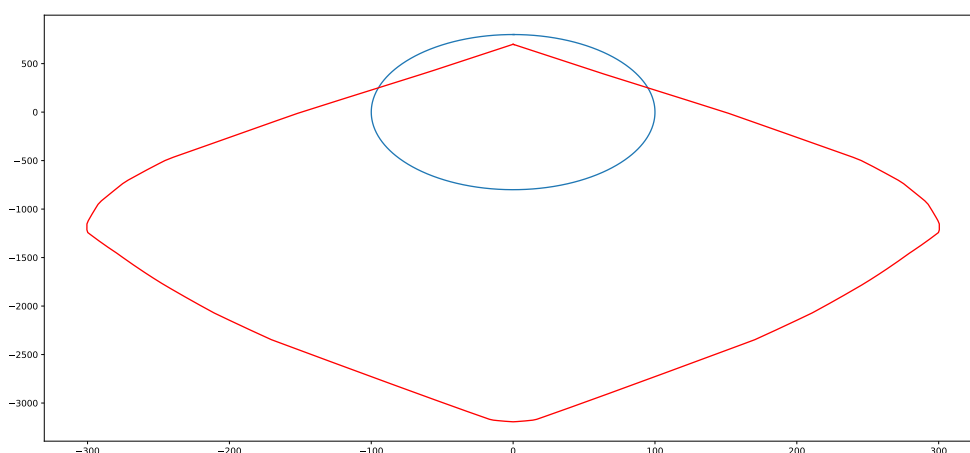


Figura 5.7: Sobreposição $SI-SR_{inicial}$

superfície resistente não contém integralmente a superfície de interação, pelo que a taxa de armadura da secção transversal terá de ser aumentada.

Com o *script* definido na Figura 5.8, o método determina o valor de $f_{\omega}^{min} = 1.335$. Conhecido o valor da taxa de armadura mínima, recorre-se ao método 3 para desenhar a superfície de resistência para o f_{ω} determinado e neste caso, a curva verde da Figura 5.9 é a superfície de resistência mínima necessária para resistir à ação sísmica. Deste modo, a Equação toma a forma de (5.11),

```
##-----MAIN-----
C3037 = Betao('C30/37', alfa=0.85)
A500 = Aco('A500')

secr = SeccaoR(0.3,0.5, C3037, A500, 10,10, arm_sup=[4,16],
              arm_inf=[4,16], arm_latE=[0,0], arm_latD=[0,0],c=0.042)
secr.met6_w(NsM=0.1)
```

Figura 5.8: *Script*

$$s^{(CQC+SRSS)} = s^{(0)} + 1 \times \frac{St}{\sqrt{St \cdot t}} \quad (5.11)$$

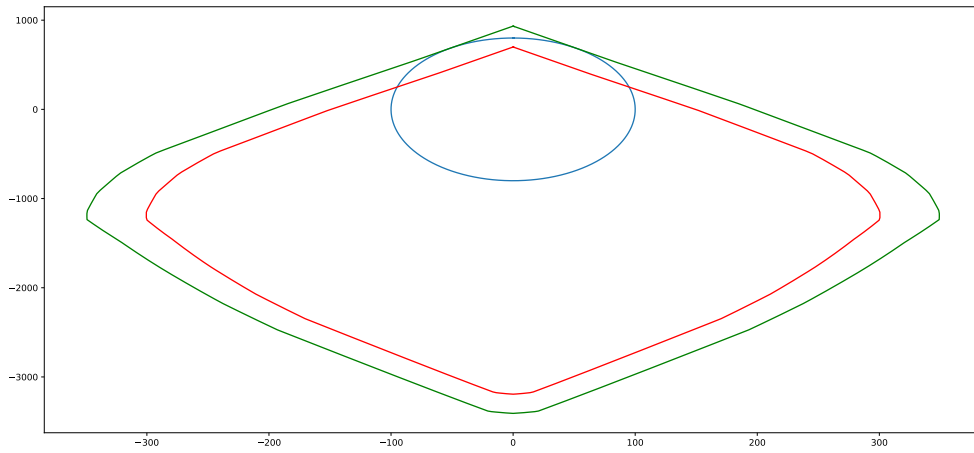


Figura 5.9: Sobreposição $SI - SR_{inicial} - SR_{ótima}$

mas para uma taxa de armadura correspondente à mínima.

Capítulo 6

Resultados

Neste capítulo, é feita uma primeira análise de curvas de interação, fazendo variar a taxa de armadura para diferentes secções. Note-se que muitas são as opções possíveis dependendo do tipo de secção e das propriedades dos materiais. Começa-se por obter as superfícies de interação resistentes para três tipos de secções transversais distintas. Seguidamente, são usados os métodos 5 e 6 para resolver os problemas de verificação de segurança e de dimensionamento da secção transversal, para uma secção transversal submetida para a ação sísmica.

6.1 Superfícies de interação

6.1.1 Secção retangular

As propriedades da secção retangular escolhida são as definidas na Tabela 6.1. A secção transversal não apresenta simetria ao nível das armaduras.

Tabela 6.1: Secção retangular

Tipo	Betão	Aço	b [m]	h [m]	c [m]	R_{inf}	R_{sup}
Retangular	C30/37	A500	0.3	0.5	0.042	5 ϕ 16	2 ϕ 16

O *script* para a criação de curvas é o ilustrado na figura 6.1 onde a secção transversal exportada está representada na Figura 6.2.

Para facilitar a sua visualização, e visto que o problema de secção composta pode ser facilmente resolvido pelo mesmo processo, as curvas de interação $N - M_y$ são ilustradas na Figura 6.3 com curvas de nível correspondentes às taxas de armadura de 0.2 em 0.2. Como era esperado, as curvas de interação não são simétricas.

Desenvolvendo o mesmo procedimento para a flexão desviada, as curvas de nível passam a ter uma componente tridimensional. Para facilitar a leitura as curvas de nível foram definidas para 4 taxas de armadura (0.5, 1.0, 1.5, 2.0), com cores distintas. Note que o utilizador tem a possibilidade de escolher apenas uma ou até mais se assim o achar necessário. As perspetiva e projeções das curvas

```

import Main as m

#-----
#Definição da secção transversal
C3037 = m.Betao('C30/37', alfa=0.85)
A500 = m.Aco('A500')
secc = m.SeccaoR(0.3,0.5, C3037, A500, 10,10, arm_sup=[2,16], arm_inf=[5,16], c=0.042)
#Curva N-My
secc.curvas_2D_desenho(0.1, 2.1, 0.2)
#Curva N-My-Mz
secc.curvas_3D_desenho(lista_taxis=[0.5, 1.0, 1.5, 2.0], NsM=10)

```

Figura 6.1: Script para modulação de curvas de interação

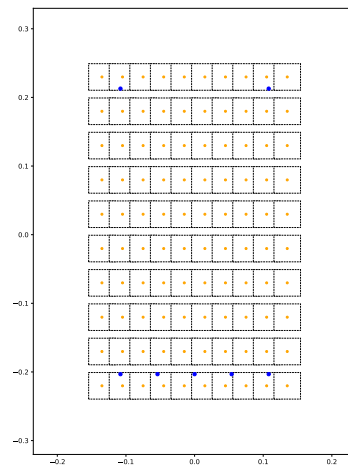


Figura 6.2: Secção retangular - discretização

de interação indicadas são ilustradas seguidamente nas Figuras 6.4 a 6.7.

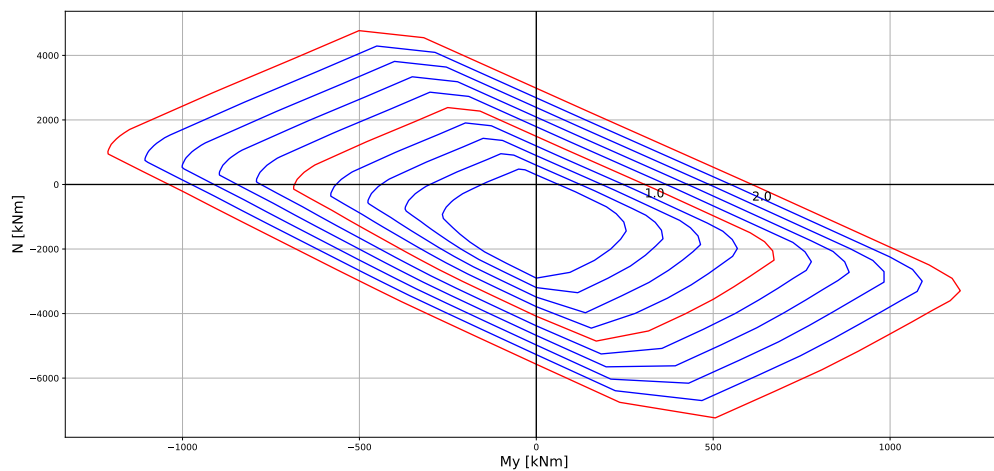


Figura 6.3: Curva de interação $N - M_y$ - Secção retangular

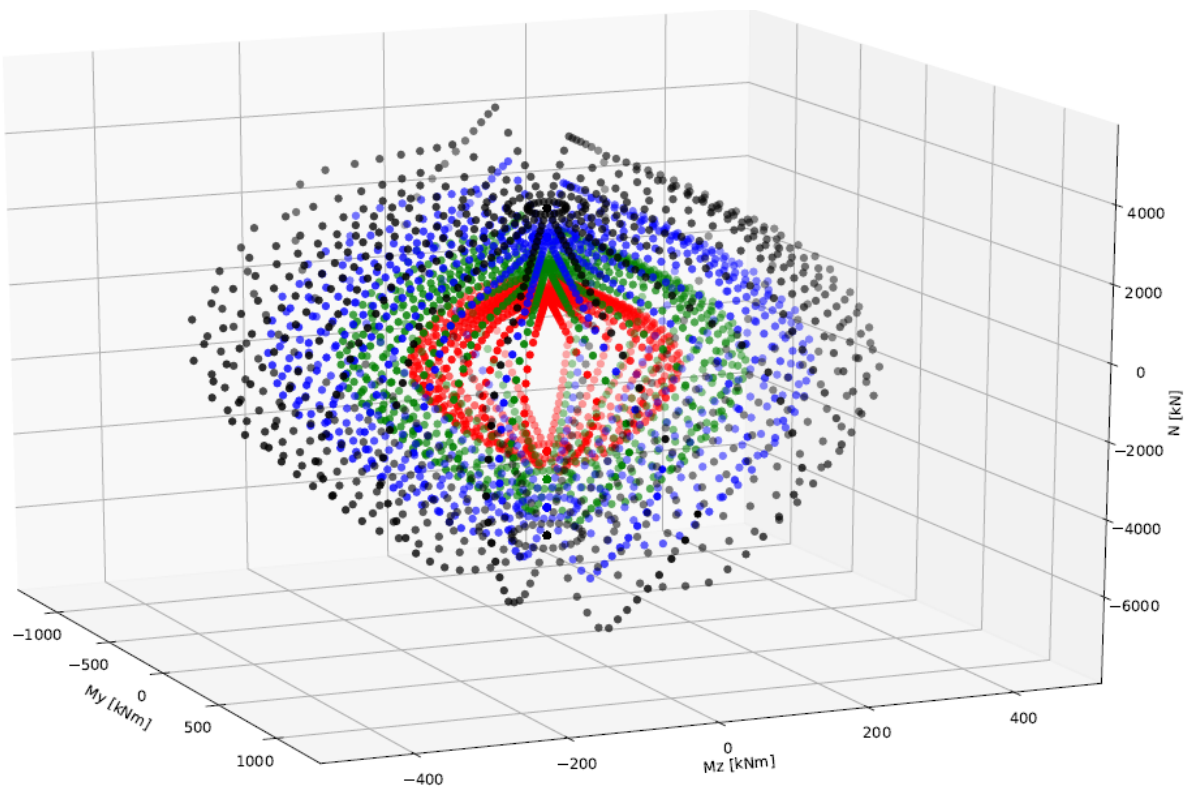


Figura 6.4: Curva de interação - Perspetiva - $N - M_y - M_z$ - Secção retangular

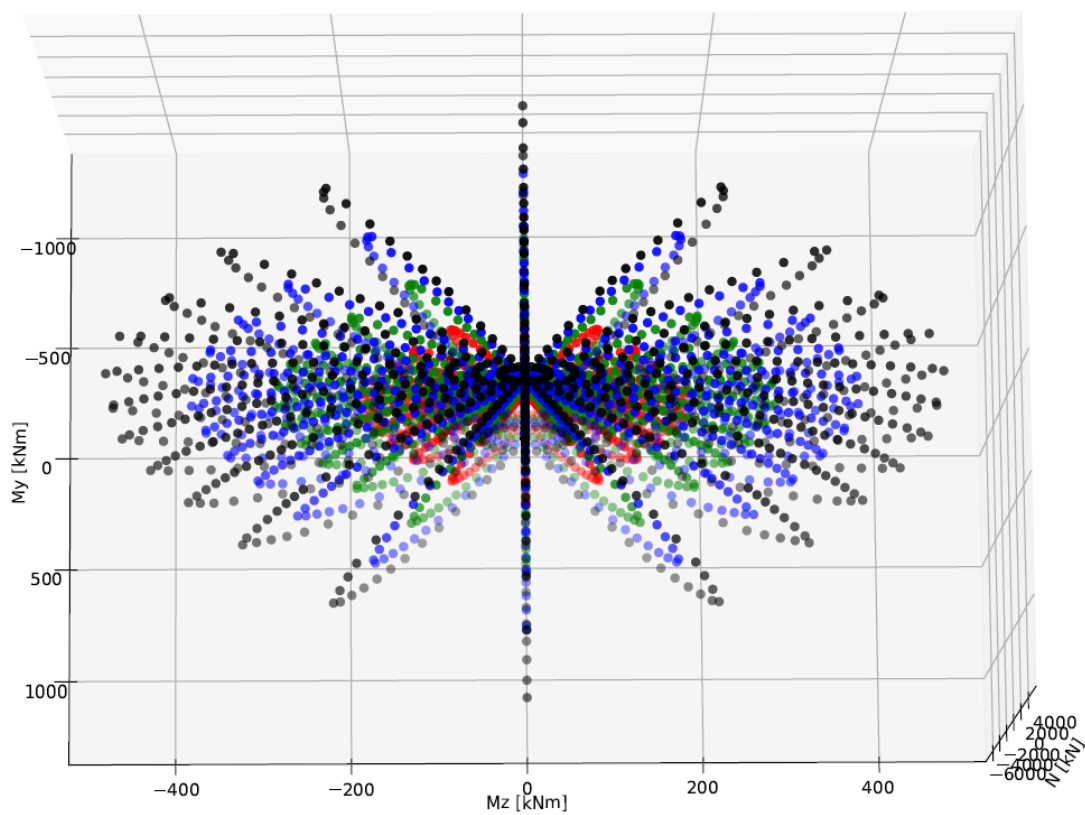


Figura 6.5: Curva de interação - projeção $M_y - M_z$ - Secção retangular

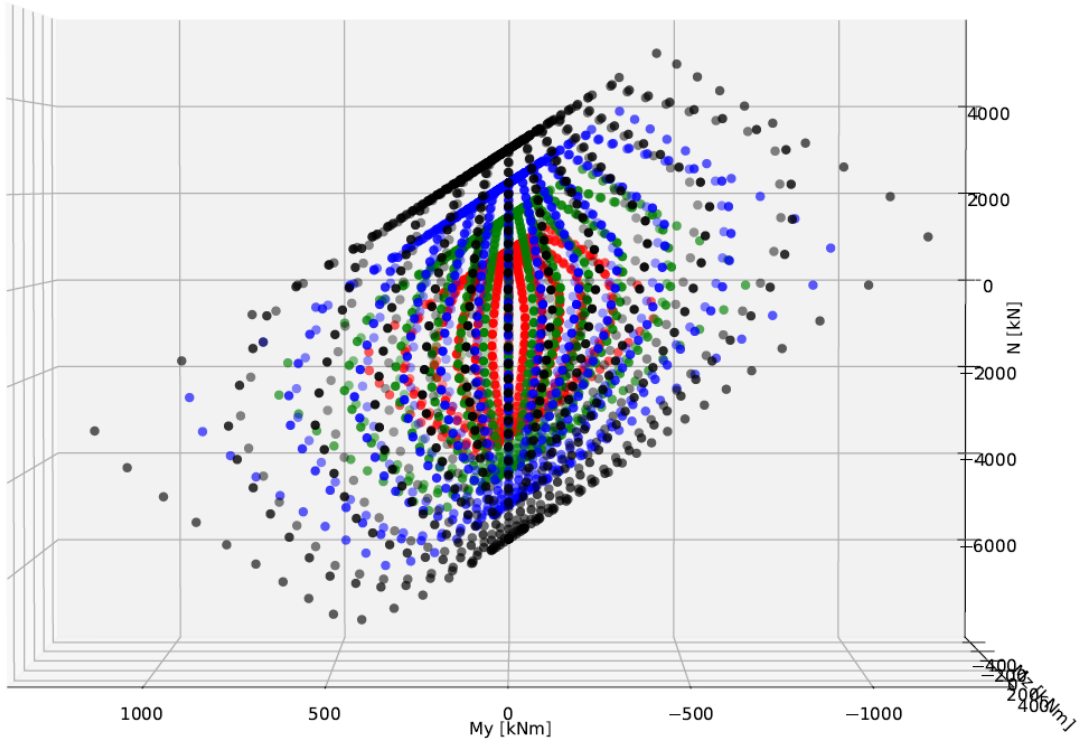


Figura 6.6: Curva de interação - projeção $N - M_y$ - Secção retangular

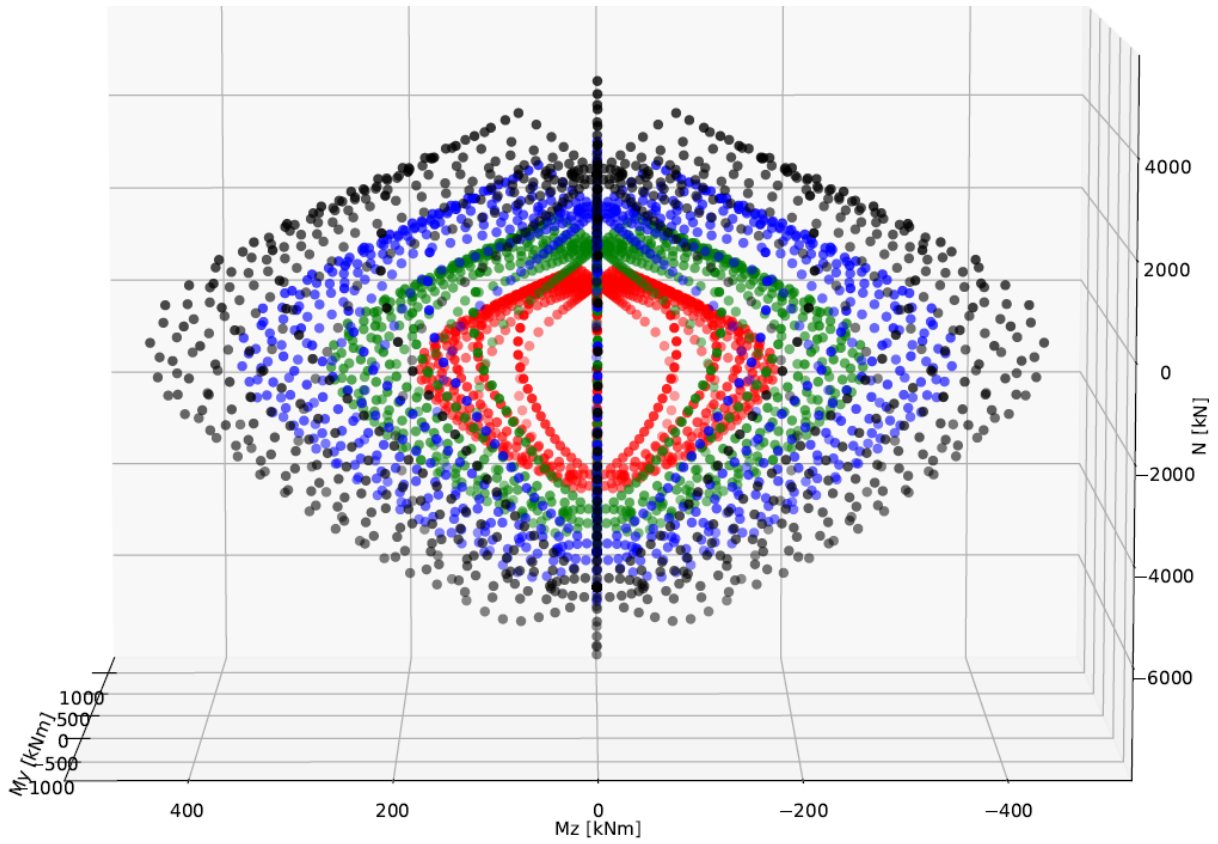


Figura 6.7: Curva de interação - projeção $N - M_z$ - Secção retangular

Existe uma distinção clara entre cada uma das projeções sendo que, não existindo simetria no eixo $y - y$, a curva de interação $N - M_y$ não é, também ela, simétrica. Para as restantes projeções, é esperada uma simetria nas curvas de interação, como evidenciado nas ilustrações. É importante referir que, para uma direção determinada, partindo de um ponto inicial, os métodos 3 e/ou 4 são os exclusivamente necessários para a verificação de segurança ou para o dimensionamento da secção transversal, respetivamente, não sendo necessário recriar toda a superfície. Para estes casos, apenas um ponto será retribuído.

6.1.2 Secção circular

As propriedades da secção circular escolhida são as definidas na Tabela 6.2. A secção apresenta simetria em relação a todos os eixos, pelo que é esperado que as curvas de interação apresentem simetria, qualquer que seja o caso.

Tabela 6.2: Secção circular

Tipo	Betão	Aço	d [m]	c [m]	R _{tot}
Circular	C20/25	A400	0.45	0.032	25 ϕ 16

O *script* para a criação de curvas é o ilustrado em 6.8 onde a secção transversal exportada está representada na Figura 6.9.

```
import Main as m

#-----
#Definição da secção transversal
C2025 = m.Betao('C20/25', alfa=0.85)
A400 = m.Aco('A400')
secc = m.SeccaoC(0.45, C2025, A400, armadura=[25,16],dr=.05, c=0.032)
#Curva N-My
secc.curvas_2D_desenho(0.1, 2.1, 0.2)
#Curva N-My-Mz
secc.curvas_3D_desenho(lista_taxas=[0.5, 1.0, 1.5, 2.0], NsM=10)
```

Figura 6.8: *Script* para modulação de curvas de interação

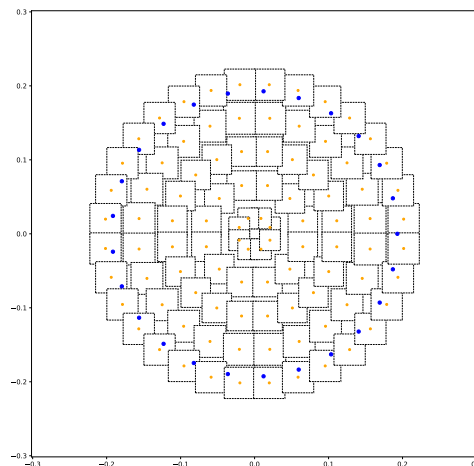


Figura 6.9: Secção circular - discretização

Para facilitar a visualização, e visto que o problema de secção composta pode ser facilmente resolvido pelo mesmo processo, as curvas de interação $N - M_y$ são ilustradas na Figura 6.3 com curvas de nível correspondentes às taxas de armadura de 0.2 em 0.2. De facto, para o caso de secções circulares simétricas, como é a maioria dos casos, a flexão desviada é facilmente transportada para um problema

de flexão composta, devido exatamente à simetria dos momentos M_y e M_z . No entanto, as superfícies tridimensionais correspondentes à flexão desviada são apresentadas para 4 taxas de armadura (0.5, 1.0, 1.5, 2.0), com cores distintas. As perspetiva e projeções das curvas de interação indicadas são ilustradas seguidamente nas Figuras 6.11 a 6.14.

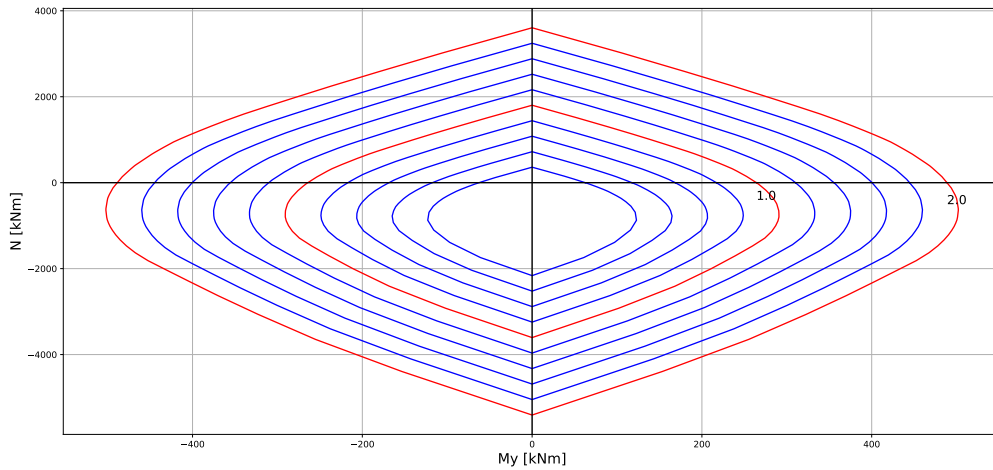


Figura 6.10: Curva de interação $N - M_y$ - Secção circular

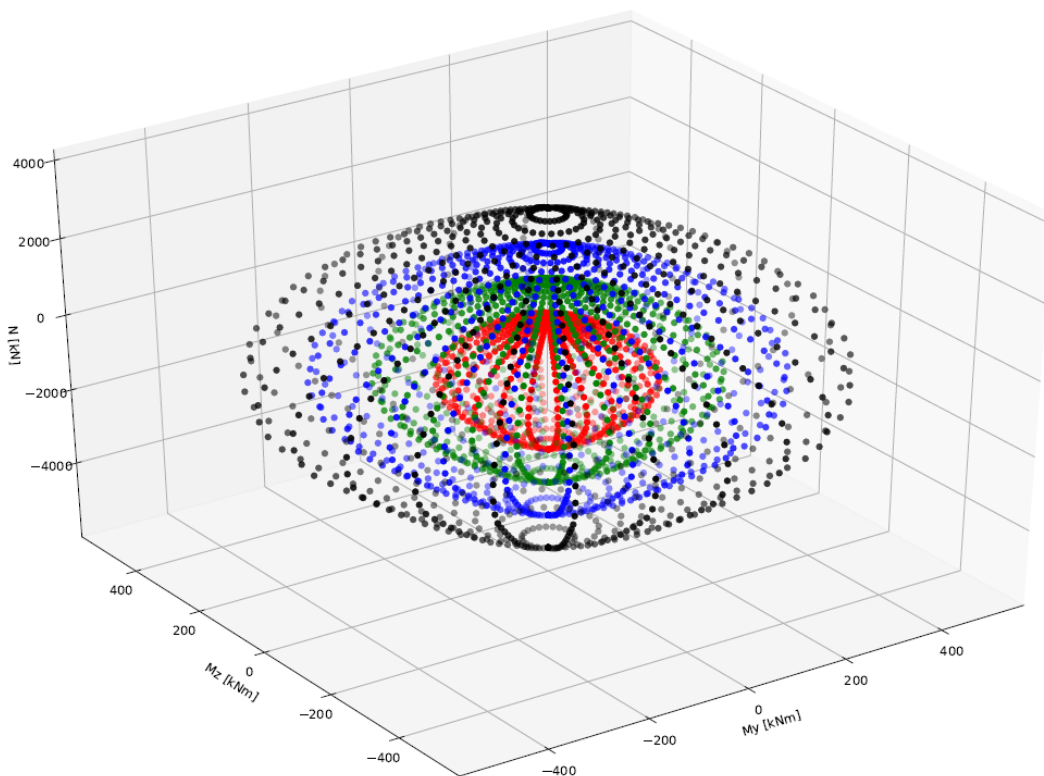


Figura 6.11: Curva de interação - Perspetiva - $N - M_y - M_z$ - Secção circular

As curvas de interação $N - M_y$ e $N - M_z$ são exatamente iguais como é possível verificar.

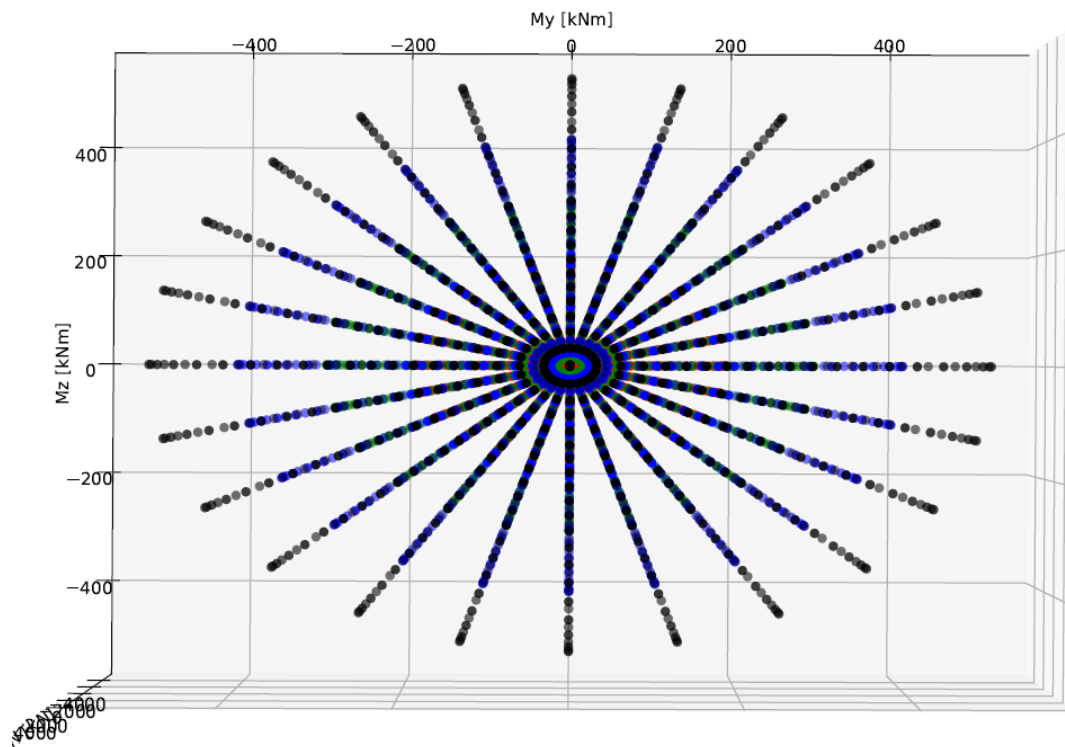


Figura 6.12: Curva de interação - projeção $M_y - M_z$ - Secção circular

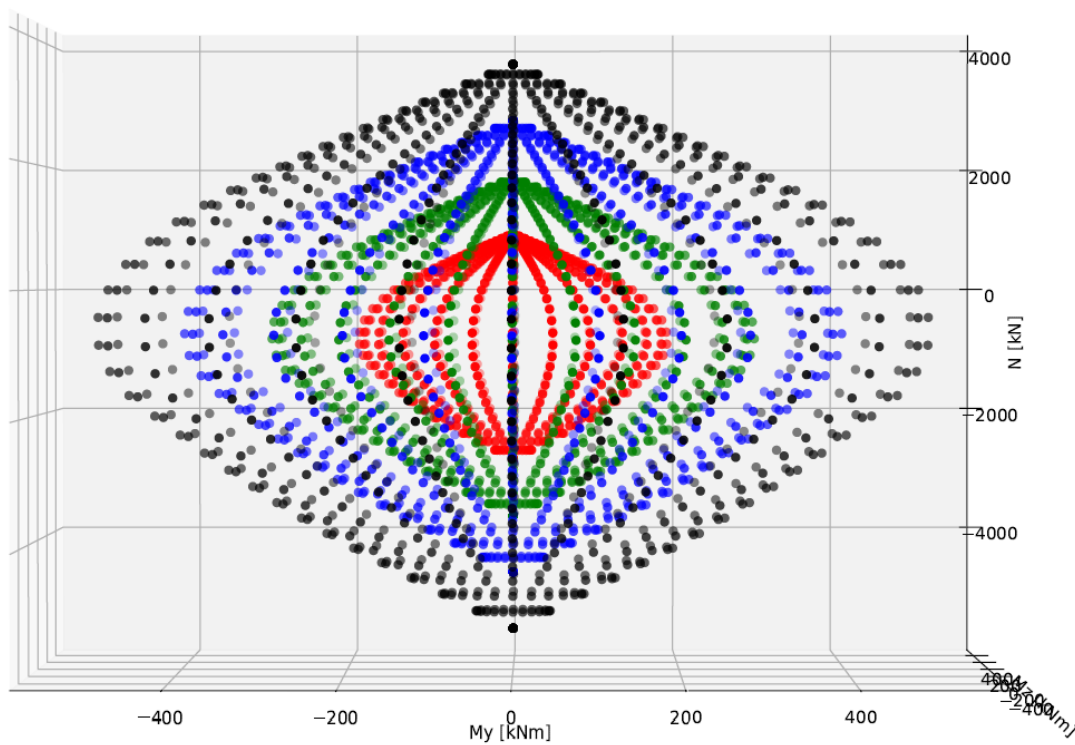


Figura 6.13: Curva de interação - projeção $N - M_y$ - Secção circular

Ainda que esta não seja a principal aplicação da dissertação, possibilita ao utilizador um pré-dimensionamento eficaz.

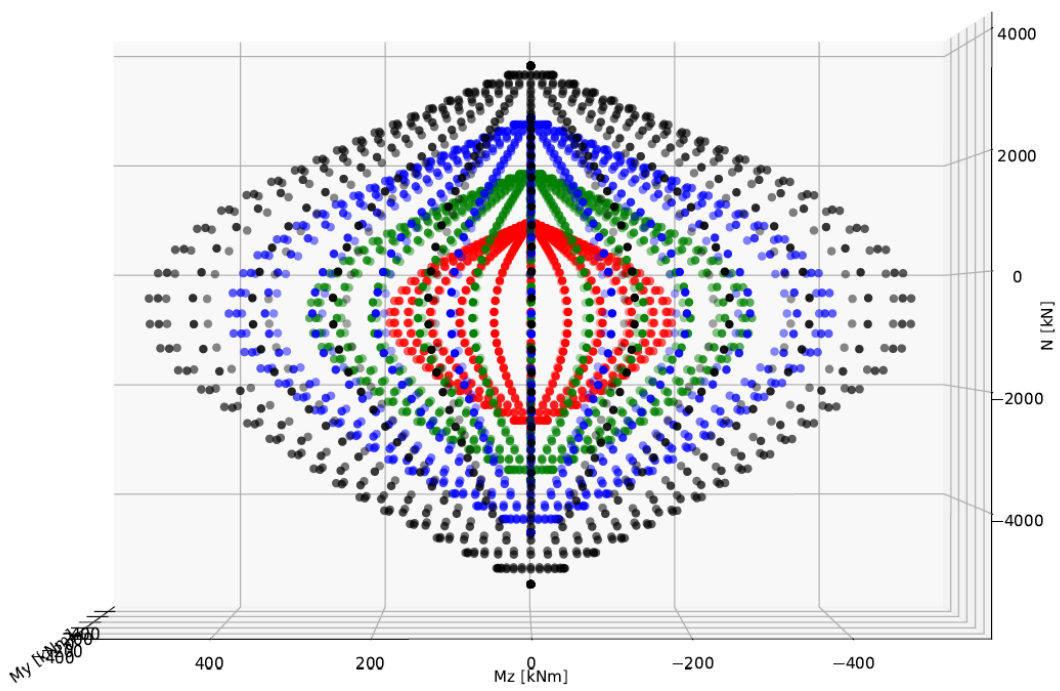


Figura 6.14: Curva de interação - projeção $N - M_z$ - Secção circular

6.1.3 Secção T

As propriedades da secção T escolhida são as definidas na Tabela 6.3. A secção transversal não apresenta simetria ao nível das armaduras para o eixo $y - y$.

Tabela 6.3: Secção T

Tipo	Betão	Aço	b_f [m]	t_f [m]	b_w [m]	t_w [m]	c [m]	R_{inf}	R_{sup}
Retangular	C35/45	A500	0.6	0.3	0.6	0.2	0.030	3 ϕ 20	(5 + 4) ϕ 16

O *script* para a criação de curvas é o ilustrado em 6.15 onde a secção transversal exportada está representada na Figura 6.16.

```
import Main as m

#-----
#Definição da secção transversal
C3545 = m.Betao('C35/45', alfa=0.85)
A500 = m.Aco('A500')
sect = m.SeccaoT(0.6,0.6,0.20,0.30, C3545, A500, ny=15,nz=15,
                arm_sup1=[5,16], arm_inf1=[4,16], arm_inf2=[3,20], c=0.03)
#Curva N-My
sect.curvas_2D_desenho(0.1, 2.1, 0.2)
#Curva N-My-Mz
sect.curvas_3D_desenho(lista_taxis=[0.5, 1.0, 1.5, 2.0], NsM=10)
```

Figura 6.15: *Script* para modulação de curvas de interação

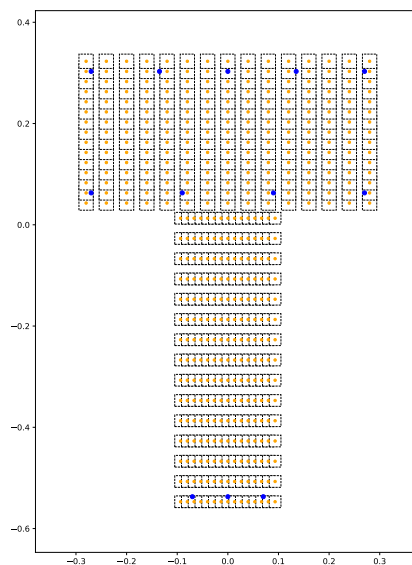


Figura 6.16: Secção T - discretização

Para facilitar a sua visualização, e visto que o problema de secção composta pode ser facilmente resolvido pelo mesmo processo, as curvas de interação $N - M_y$ são ilustradas na Figura 6.17 com

curvas de nível correspondentes às taxas de armadura de 0.2 em 0.2. Como era esperado, as curvas de interação não são simétricas.

Desenvolvendo o mesmo procedimento para a flexão desviada, as curvas de nível passam a ter uma componente tridimensional. Para facilitar a leitura, as curvas de nível foram definidas para 4 taxas de armadura (0.5, 1.0, 1.5, 2.0). Os resultados são apresentados nas Figuras 6.18 a 6.21.

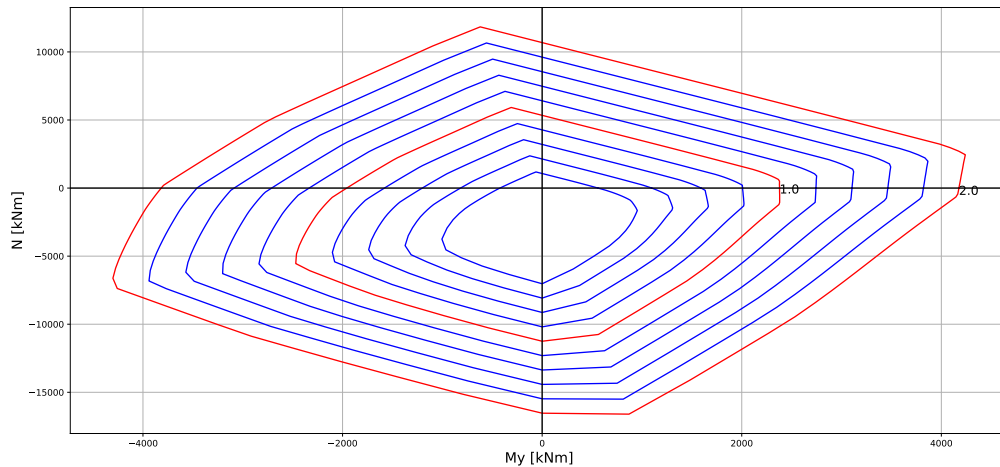


Figura 6.17: Curva de interação $N - M_y$ - Secção T

A assimetria da secção é notável na projeção $N - M_y$, como seria de esperar.

São inúmeras as secções que se podem determinar, de uma forma simples, reforçando que para o caso de se conhecer a direção de esforços, só será necessário definir um ponto da superfície de resistência da secção transversal.

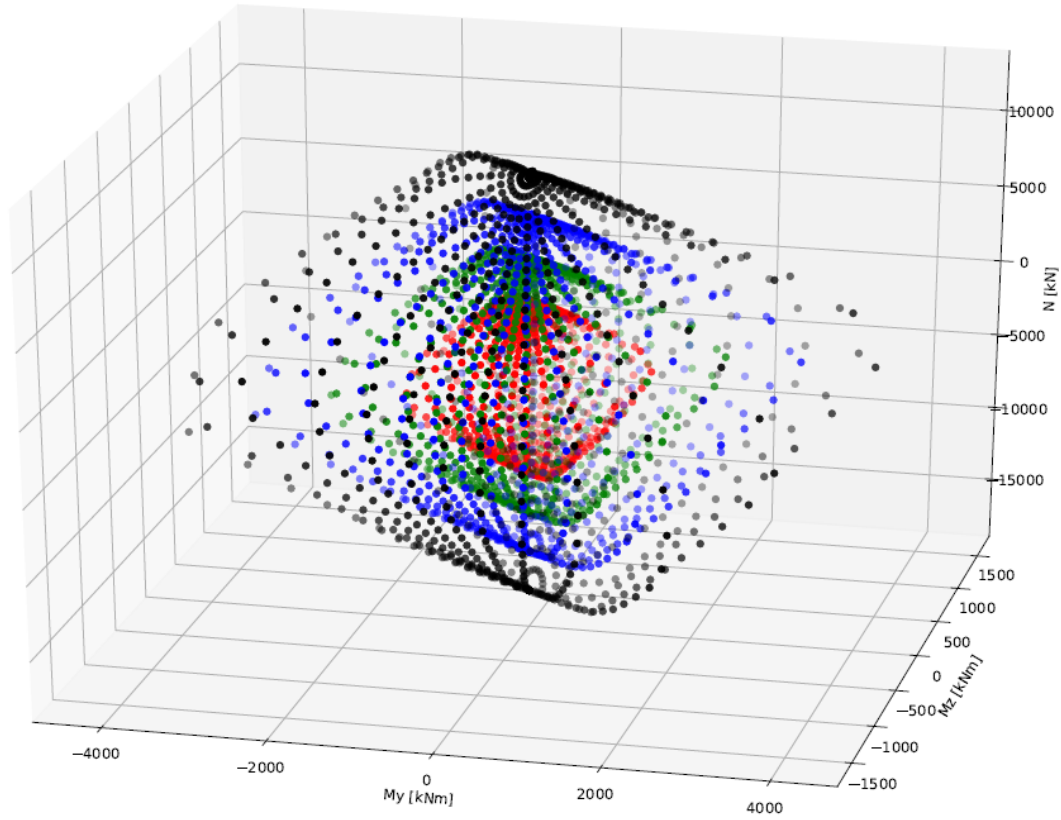


Figura 6.18: Curva de interação - Perspetiva - $N - M_y - M_z$ - Secção T

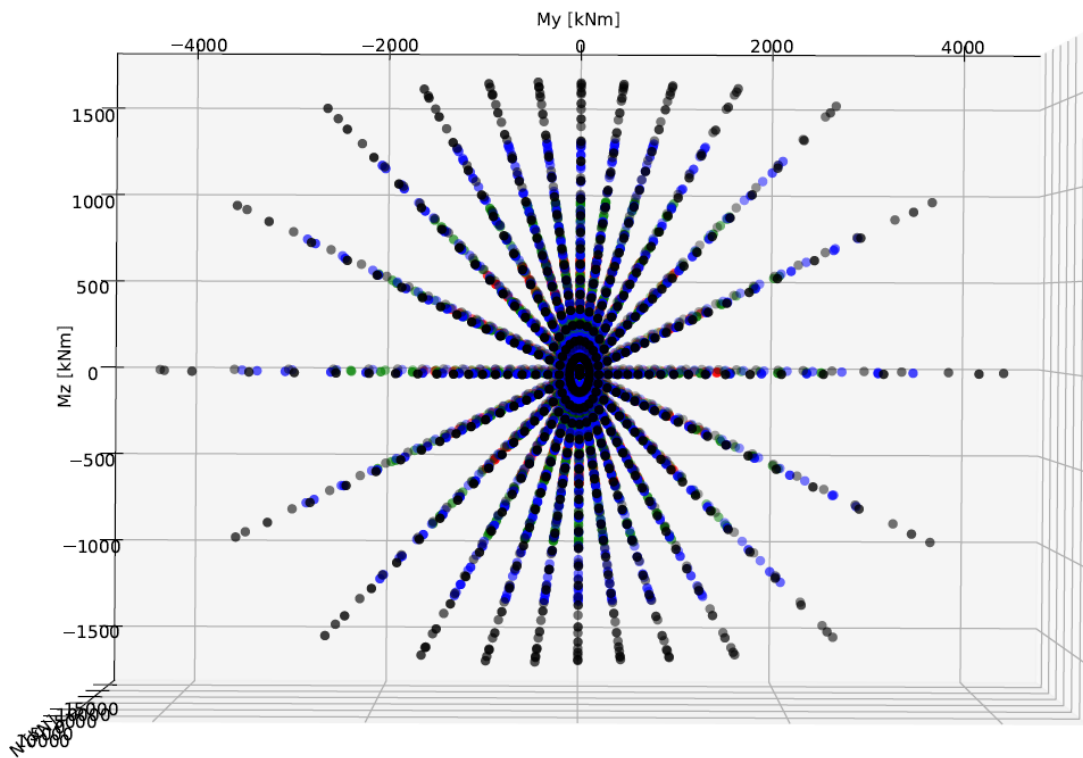


Figura 6.19: Curva de interação - projeção $M_y - M_z$ - Secção T

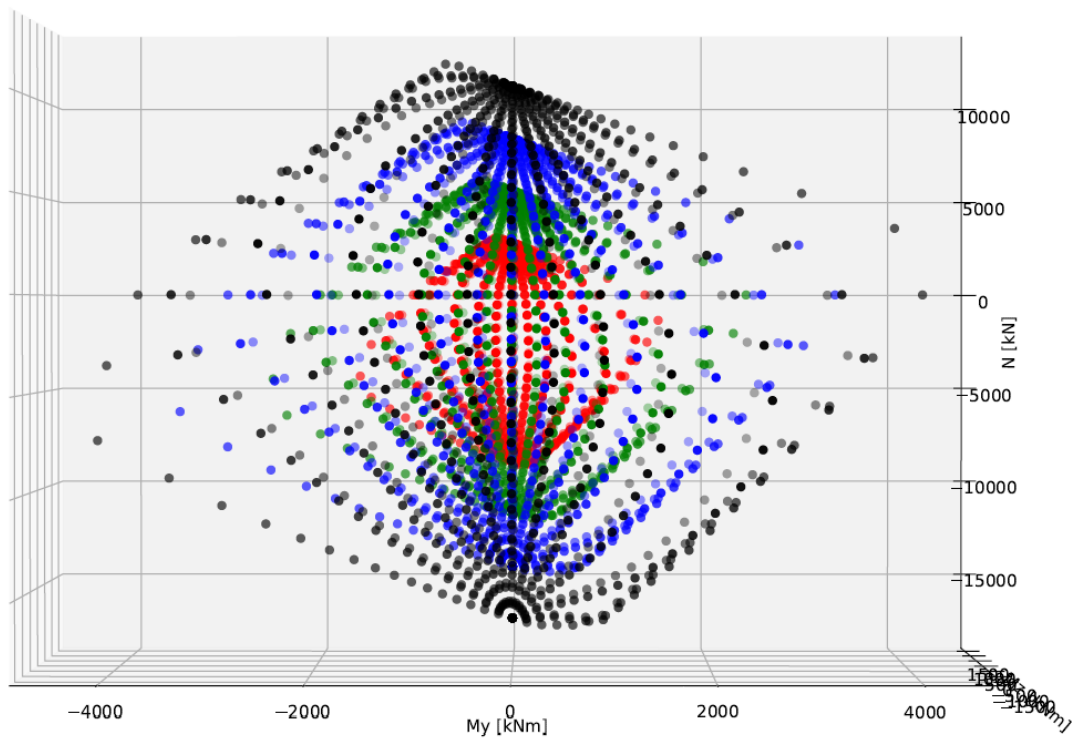


Figura 6.20: Curva de interação - projeção $N - M_y$ - Secção T

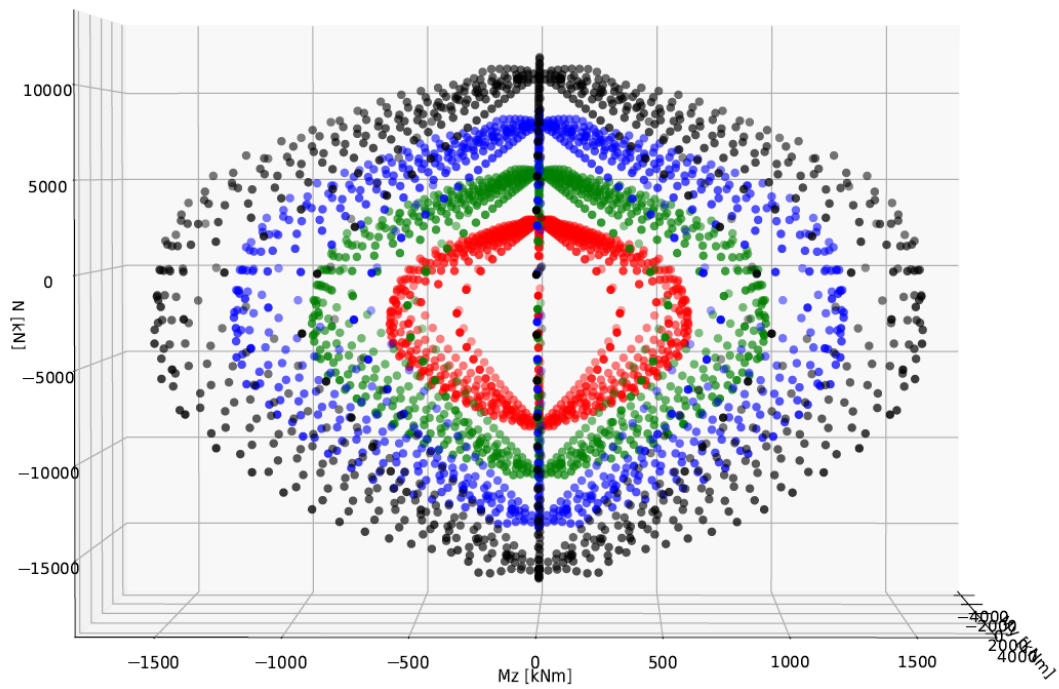


Figura 6.21: Curva de interação - projeção $N - M_z$ - Secção T

6.2 Superfícies de resistência para uma análise sísmica

Chegando à secção principal de toda a dissertação, o resultado que se segue considera as duas possíveis situações para a ocorrência de uma ação sísmica: verificação de segurança e o seu dimensionamento. Seja a ação1, definida pelas suas componentes permanentes $s^{(0)}$ e dinâmicas S em (6.1), a ação sísmica a atuar na secção transversal retangular definida na Tabela 6.4 e ilustrada na Figura 6.23, através do *script* definido em 6.22.

$$s^{(0)} = \begin{bmatrix} -150 & 50 & 0 \end{bmatrix} \quad S = \begin{bmatrix} 640100 & -24000 & 500 \\ -24000 & 23400 & 0 \\ 500 & 0 & 2500 \end{bmatrix} \quad (6.1)$$

Tabela 6.4: Secção retangular

Tipo	Betão	Aço	b [m]	h [m]	c [m]	R_{inf}	R_{sup}
Retangular	C25/30	A500	0.3	0.3	0.04	4 ϕ 16	4 ϕ 16

```
C3037 = Betao('C30/37', alfa=0.85)
A500 = Aco('A500')
seccr = SeccaoR(0.3,0.3, C3037, A500, ny=10,nz=10,
               arm_sup=[4,16], arm_inf=[4,16], c=0.04)
```

Figura 6.22: Script da secção transversal

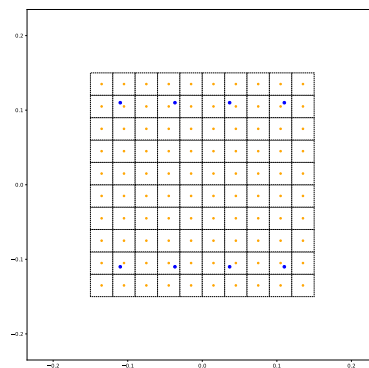


Figura 6.23: Secção transversal

6.2.1 Ampliação da componente dinâmica da ação sísmica

Correndo o *script* definido na Figura 6.24 obtêm-se as Figuras 6.25 a 6.31. Na representação tridimensional a curva de interação da ação sísmica inicial é a representada a vermelho. A superfície resistente não envolve completamente a superfície de interação, o que significa que existe um fator de segurança

para a ampliação do sismo inferior a 1. O fator λ é igual a 0.291, ou seja, a componente dinâmica do sismo teria de diminuir até 0.291 vezes para atingir a superfície resistente da secção transversal. A curva de interação para a qual a secção transversal resiste é a representada a azul.

```
secr.met5_lambda(NsM=10)
secr.met5_lambda_3D(NsM=10)
```

Figura 6.24: Script para o método 5

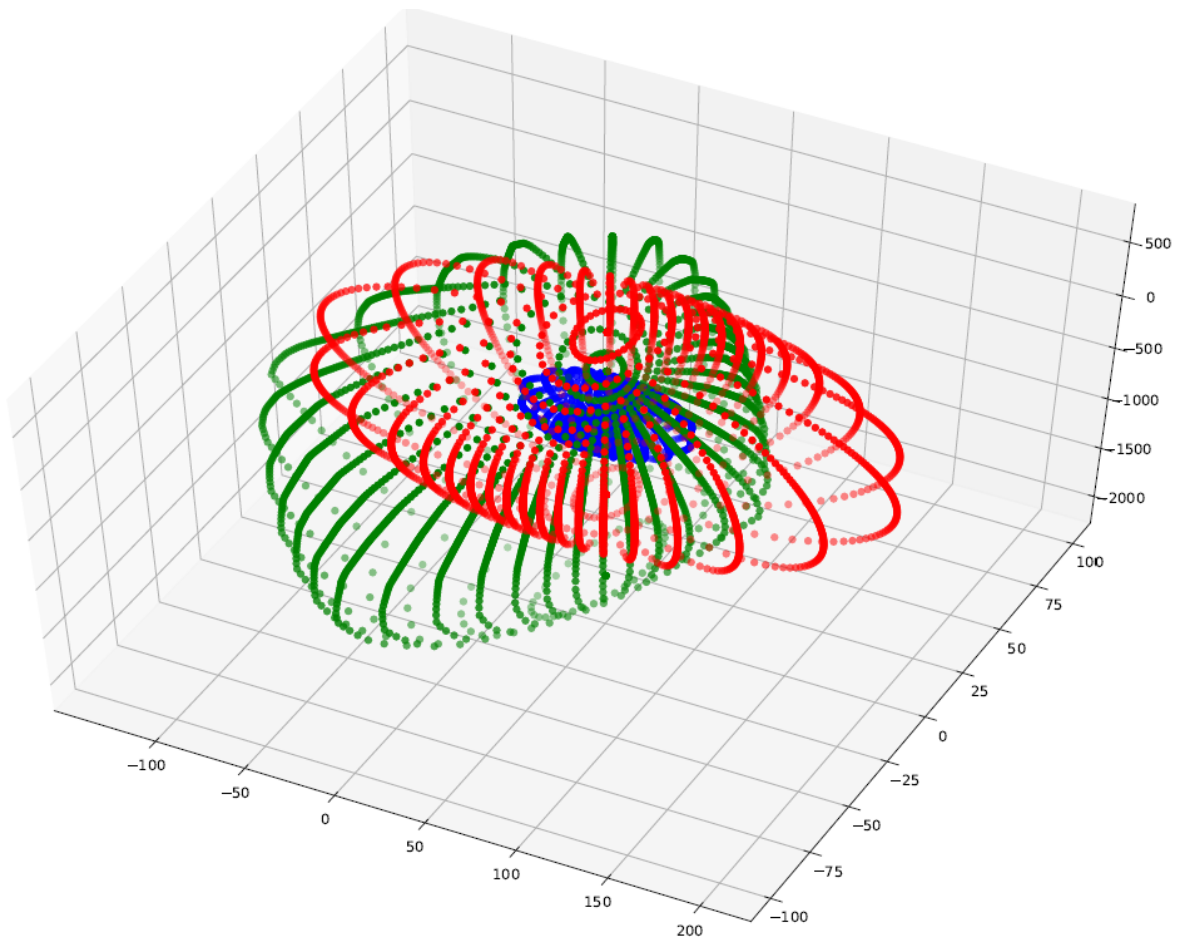


Figura 6.25: Perspetiva 3D (1)

Para uma melhor visualização projetam-se as curvas para uma representação bidimensional. Sendo que a azul é representada a curva de interação da ação sísmica e a vermelho a curva de resistência da secção transversal, é agora visualmente acessível entender que a secção transversal não resistirá à ação definida pelo sismo. A curva a verde é uma diminuição da inicial através do fator $\lambda = 0.291$ com centro em $s^{(0)} = [N = -150kN, M_y = 50kNm, M_z = 0kNm]$. Note que o centro da superfície de interação que representa a componente permanente das ações exercidas, $s^{(0)}$, não estando centrado com o centro da curva de interação resistente faz com que a componente dinâmica do sismo tenha de ser inferior.

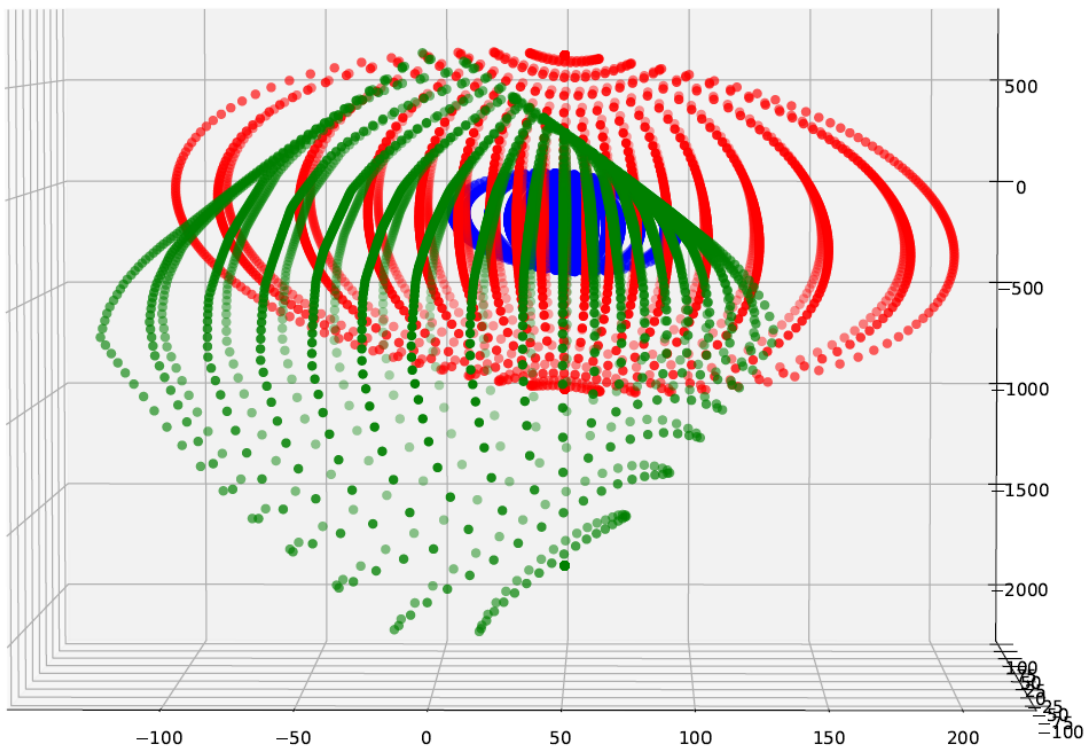


Figura 6.26: Perspetiva 3D (2)

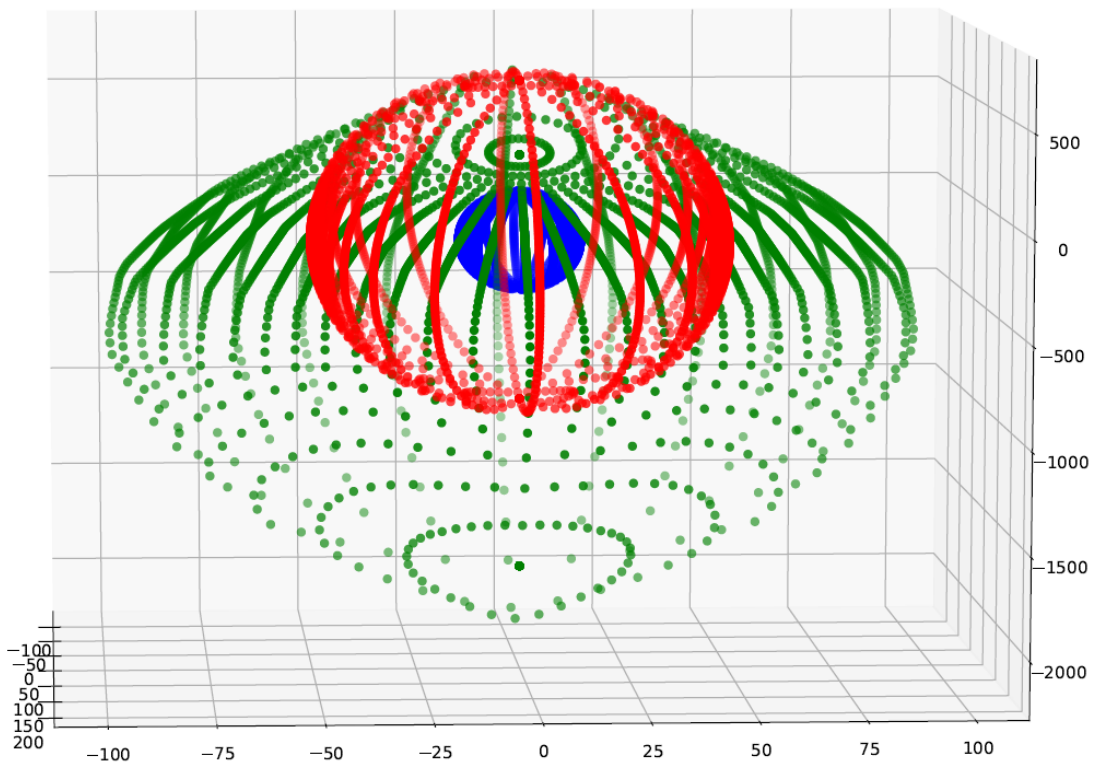


Figura 6.27: Perspetiva 3D (3)

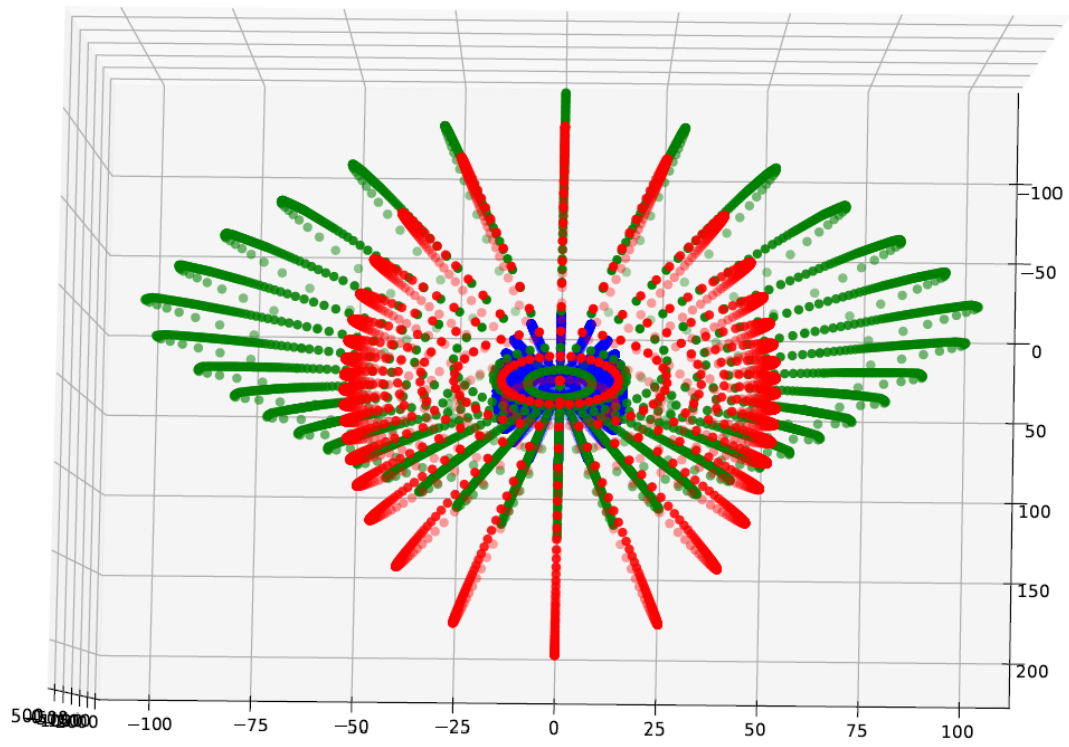


Figura 6.28: Perspetiva 3D (4)

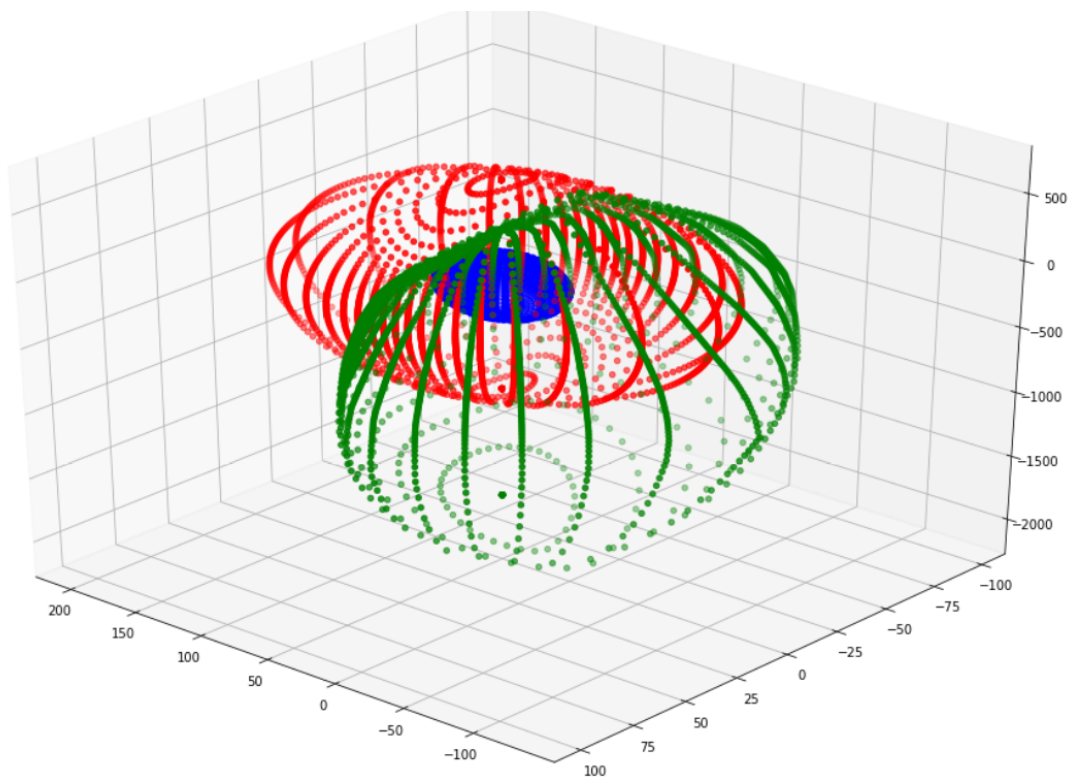


Figura 6.29: Perspetiva 3D (5)

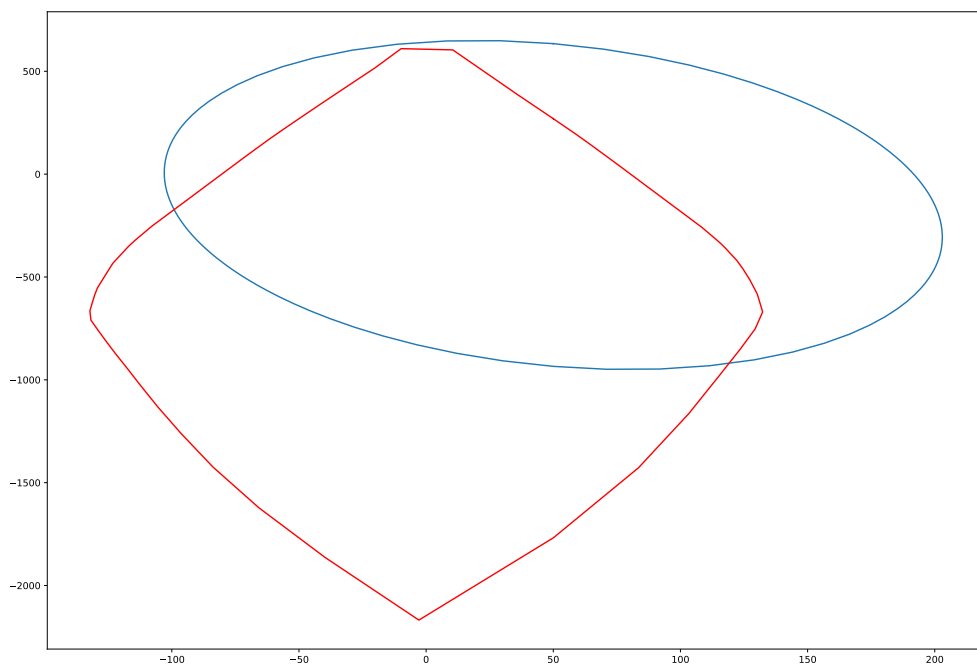


Figura 6.30: Superfície de interação e resistência -Projeção 2D

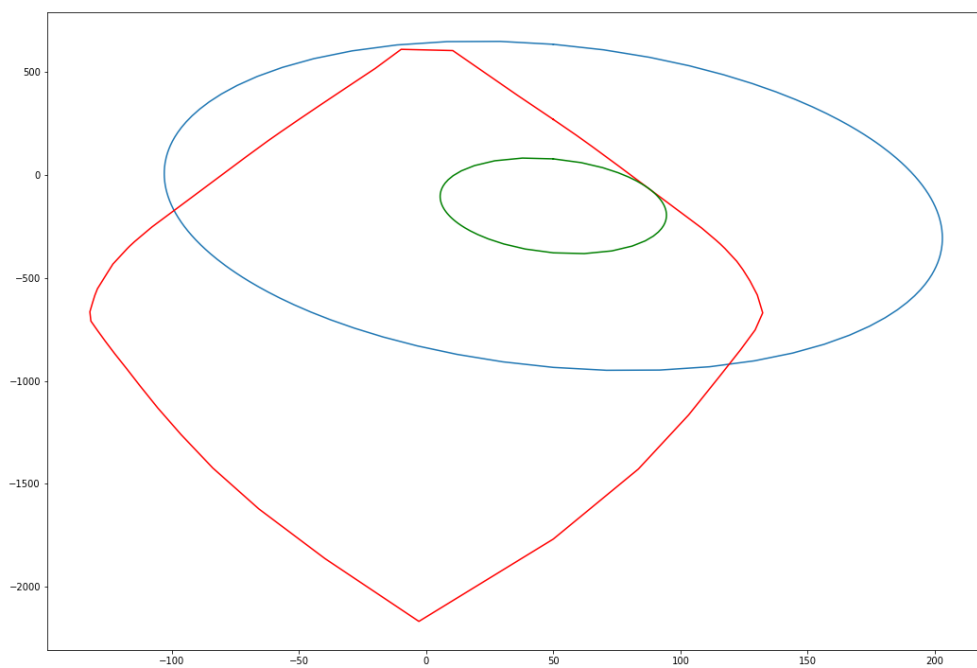


Figura 6.31: Superfície de interação final e resistência -Projeção 2D

6.2.2 Otimização da taxa de armadura para uma ação sísmica

Correndo o *script* definido na Figura 6.32 obtêm-se as Figuras 6.33 a 6.38. Na representação tridimensional a curva de interação da ação sísmica é a representada a vermelho. A superfície resistente, representada a verde, não envolve completamente a superfície de interação, o que significa que é necessário um aumento de armadura para que a secção transversal possa resistir à ação sísmica. O fator f_w é igual a 2.479, ou seja, a armadura da secção transversal terá de ser aumentada em 2.479 vezes para que possa resistir às ações atuantes. A superfície resistente mínima para resistir à ação sísmica é a representada a azul.

```
secr.met6_w(NsM=10)  
secr.met6_w_3D(NsM=10)
```

Figura 6.32: Script para método 6

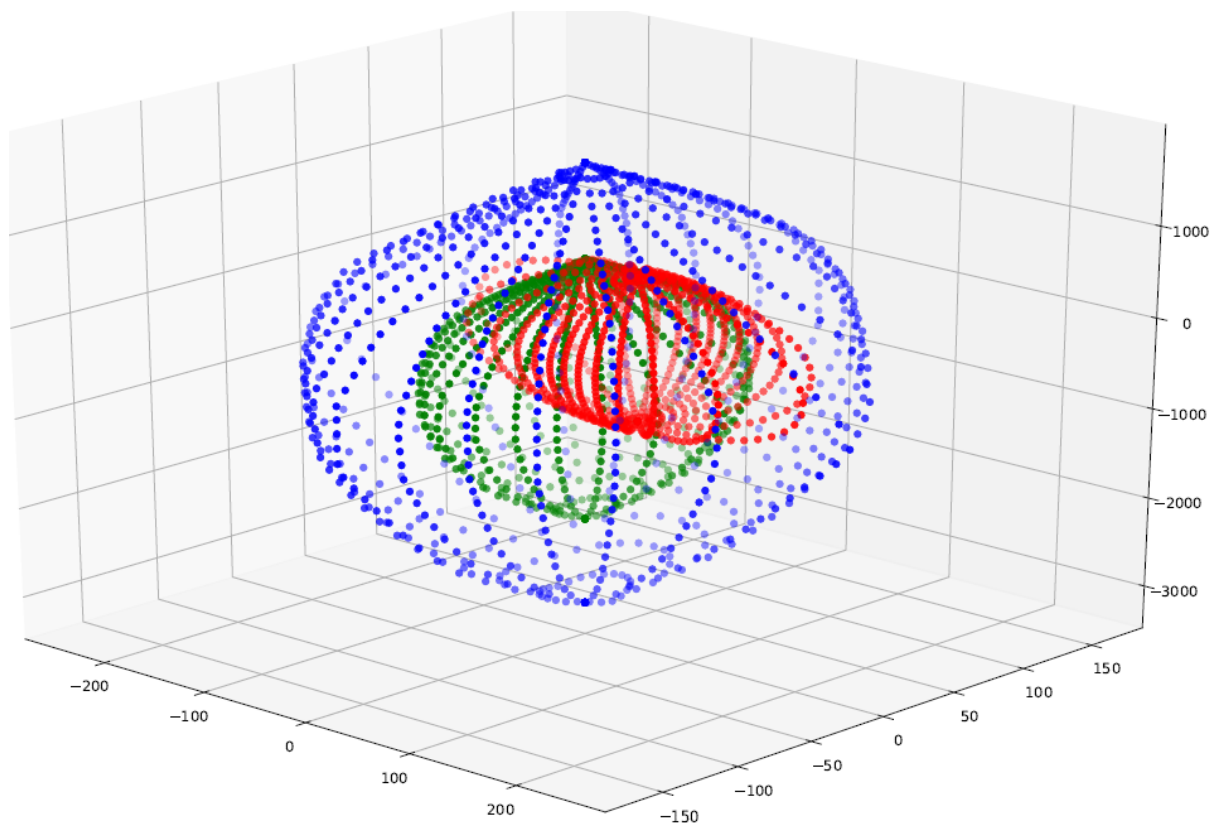


Figura 6.33: Perspetiva 3D (1)

Projetando as curvas para uma representação bidimensional, sendo que a azul é representada a curva de interação da ação sísmica, a vermelho a curva de resistência da secção transversal inicial e a verde a superfície resistente otimizada, é agora visualmente acessível entender que a secção transversal inicial não resistiria à ação dinâmica definida pelo sismo.

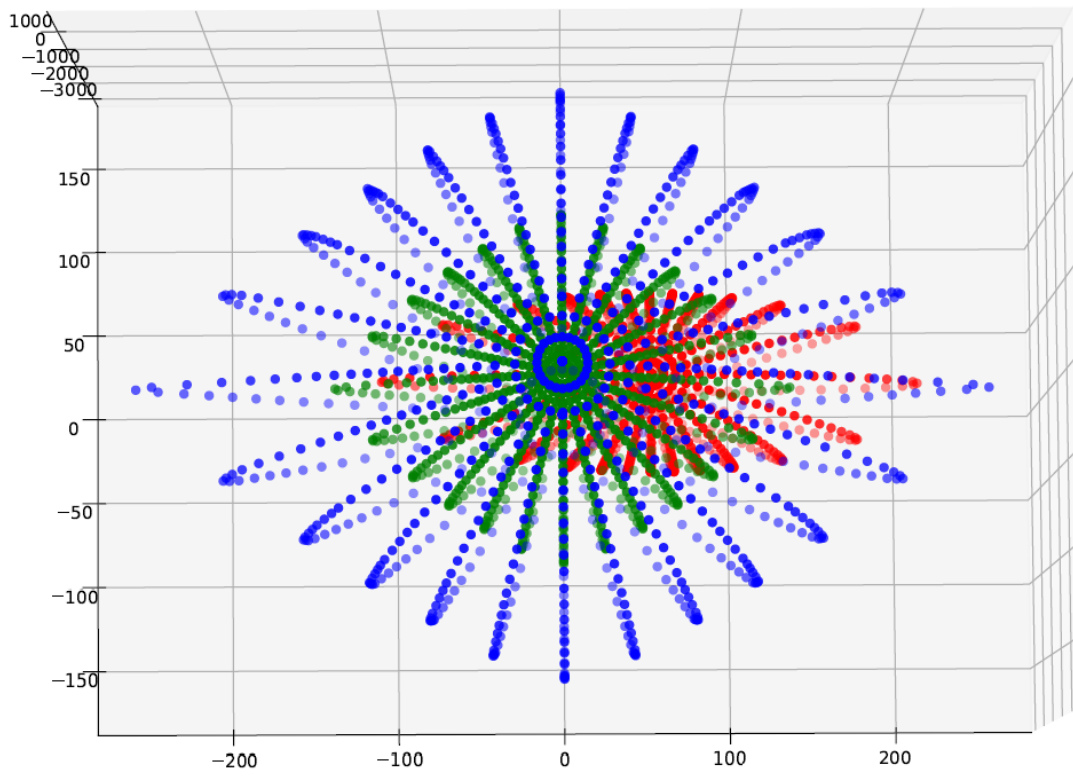


Figura 6.34: Perspetiva 3D (2)

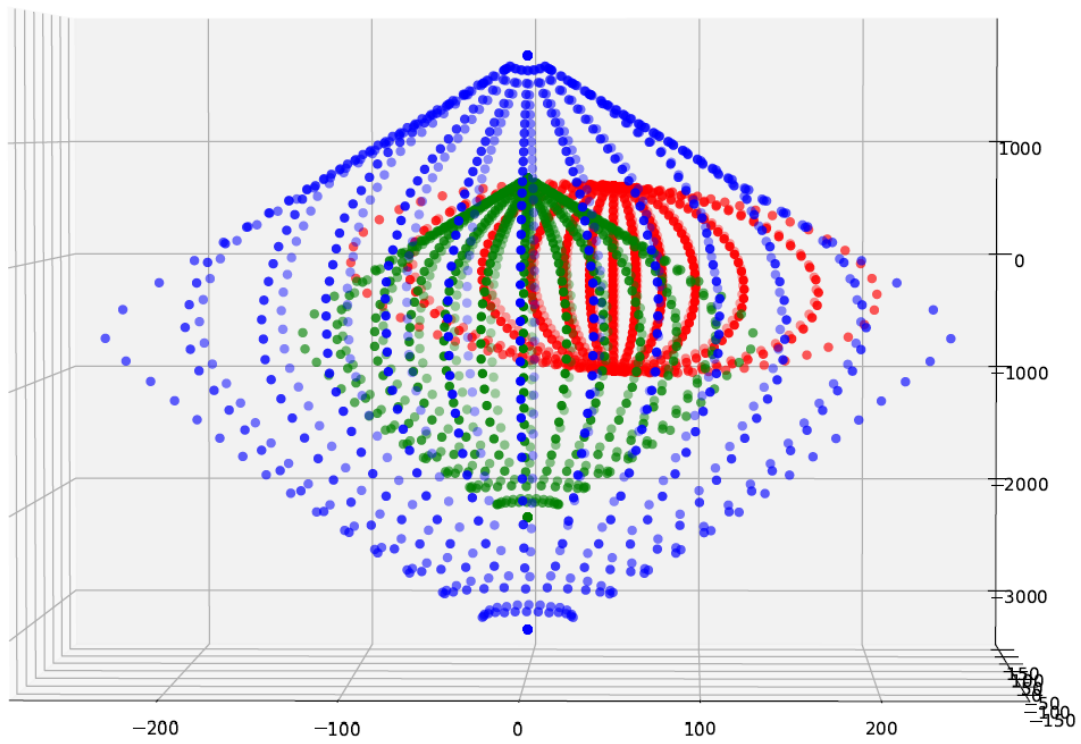


Figura 6.35: Perspetiva 3D (3)

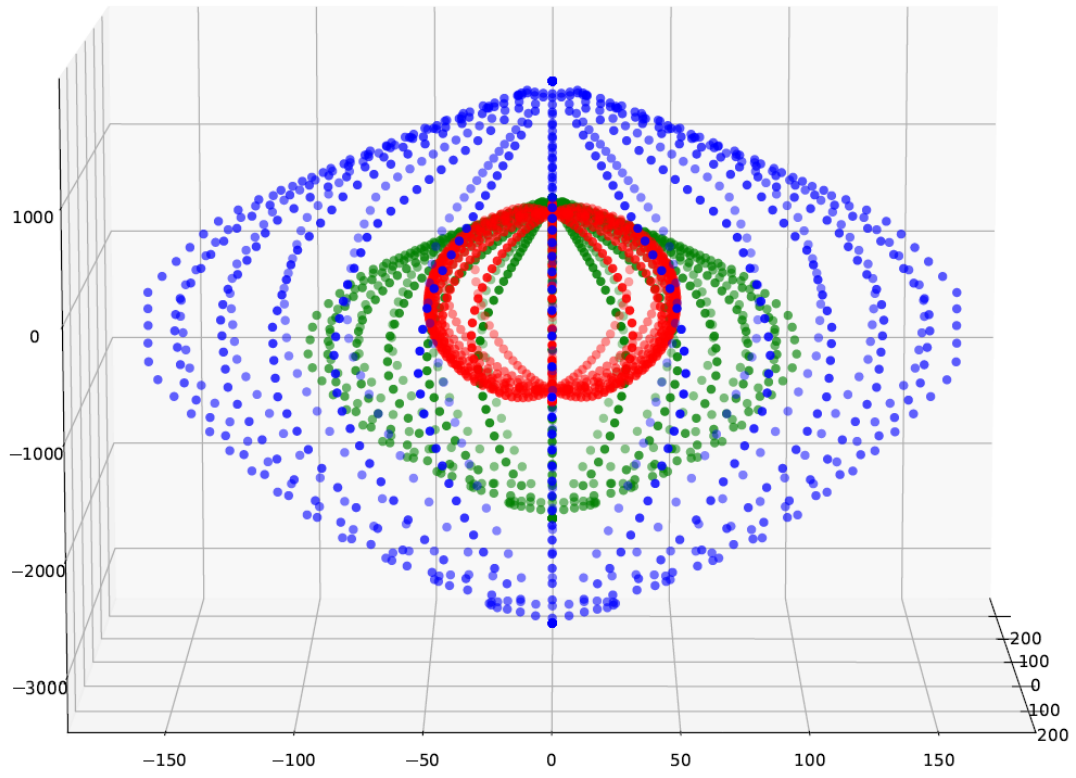


Figura 6.36: Perspetiva 3D (4)

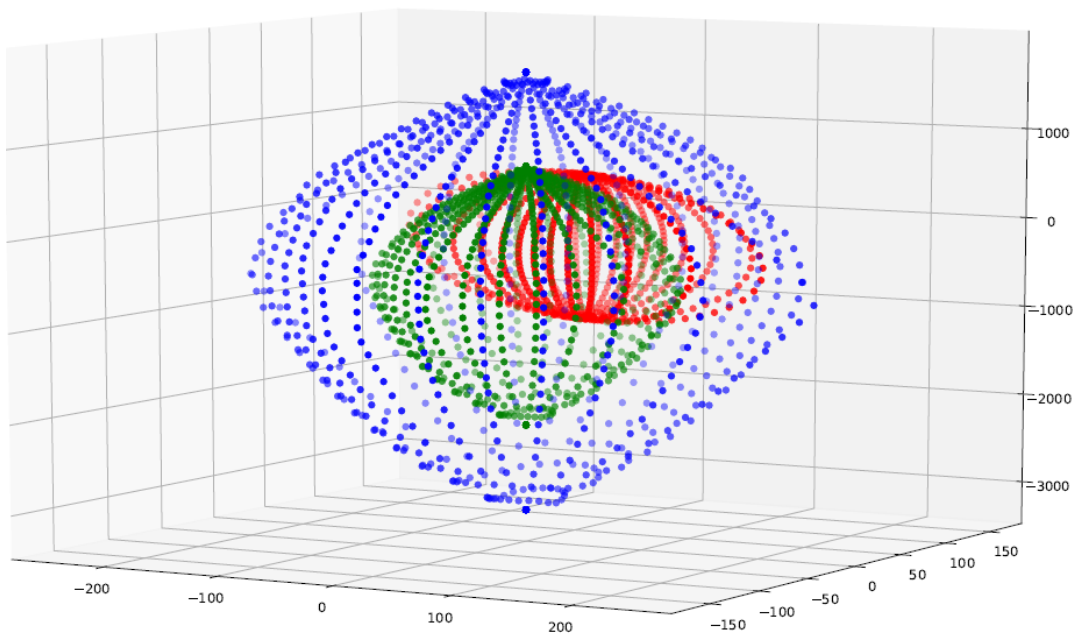


Figura 6.37: Perspetiva 3D (5)

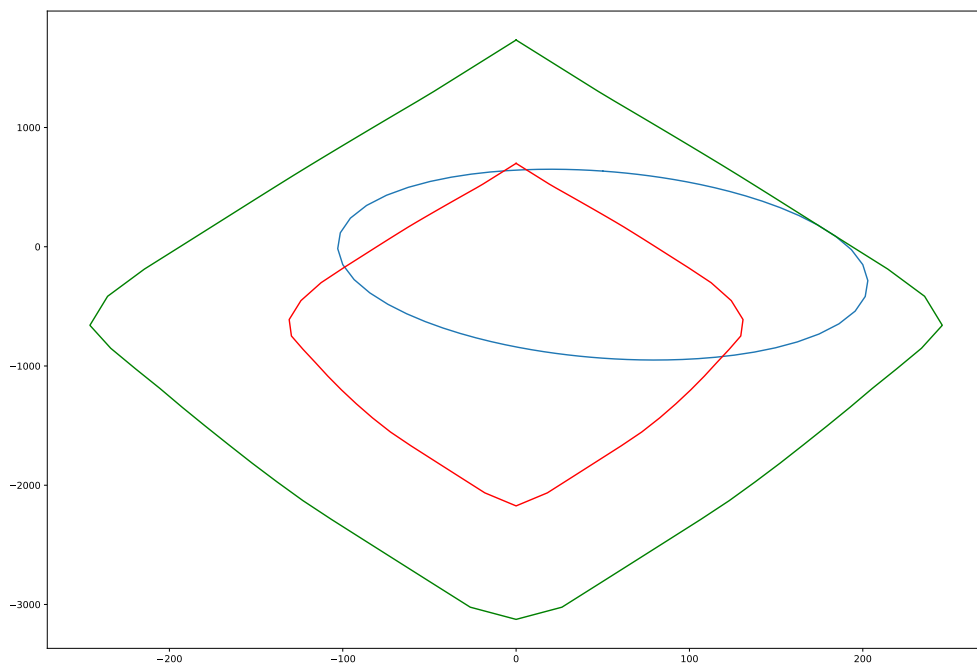


Figura 6.38: Superfície de interação e resistência final -Projeção 2D

Capítulo 7

Conclusões

Sendo o principal objetivo desta dissertação definir superfícies de resistência mínimas que resistam à totalidade da envolvente de esforços atuantes, foram criados vários métodos iterativos. De um modo geral, o objetivo final do presente trabalho é construir uma superfície resistente que tenha um ponto tangente à superfície de interação, seja ela qual for, e que todos os outros pontos da superfície resistente envolvam a superfície de interação, isto é, a superfície de resistência deverá ser a mínima possível que envolva completamente a superfície de interação. De outro modo, a superfície de interação terá de estar completamente contida no domínio da superfície de resistência.

Os primeiros métodos, métodos 1 e 2, sendo os já conhecidos e mais estudados, serviram de base à construção dos métodos seguintes. O método 3 pode ser considerado numa verificação de segurança, uma vez que dada uma direção de esforços e um ponto inicial, retribui o valor pelo qual a direção fornecida pode ser multiplicada até ser atingida a superfície de interação, o que significa que se a direção for conhecida, será somente necessário definir um ponto da secção resistente. Já o método 4 deve ser considerado num dimensionamento. Ao contrário do método 3, o método 4 define a taxa de armadura mínima para resistir à combinação de esforços atuantes fornecidas, neste caso com uma direção conhecida. Se essa direção for totalmente desconhecida será necessário pensar na superfície resistente em vez de um ponto. Nesse caso, os métodos 3 e 4 são extrapolados para os métodos 5 e 6, criando rotinas para o reconhecimento de toda a superfície.

A programação desenvolvida nesta tese permite também definir curvas de interação para qualquer tipo de secção transversal definidas no trabalho, ou qualquer conjugação das mesmas, vantagem em relação aos ábacos das tabelas de betão armado, uma vez que estão limitados a certo tipo de secções e recobrimentos, e a sua precisão é inferior. Os ábacos foram definidos com uma redução de 15% no valor de rotura do betão à compressão, e como já foi explicado anteriormente, o EC2 permite que não exista essa redução para situações correntes, uma vez que nas condições de carregamentos em tempo prolongado o betão está, em geral, solicitado a menores níveis de tensões. A programação orientada por objetos permite também que, após ser criado o programa principal, cada método possa ser definido com um número bastante reduzido de linhas de código, tornando acessível a sua criação. Teoricamente, deu-se mais um passo em frente no desenvolvimento desta limitação, mas é importante

criar bases sólidas para a implementação de um dimensionamento baseado em superfícies de interação que, não sendo sobredimensionadas, não comprometam a segurança estrutural.

Desenvolvimentos futuros

Considerando os princípios anteriores, e tendo por base a programação já desenvolvida na presente dissertação, propõe-se o desenvolvimento de métodos de convergência no que diz respeito às superfícies de interação. Acredita-se que é possível obter a tangência de ambas as superfícies, atuantes e resistentes, sem que seja necessário um número vasto de iterações.

A extensão do programa para estados limites de serviço é também uma abordagem interessante ao problema, sendo que grande parte do tempo de vida útil de uma estrutura se encontra nesta fase.

Propõe-se também a extensão do programa para estruturas metálicas. Muito embora o programa seja versátil do ponto de vista dos materiais, certas considerações devem ser tidas em conta, como é o caso da encurvadura lateral do aço sujeito a compressões, especialmente para estruturas de classe 4.

Bibliografia

- [1] CEN, 2004. Eurocode 2: Design of concrete structures - Part 1-1: General rules and rules for buildings. *European Standard EN 1992-1-1*.
- [2] CEN, 2005. Eurocode 8-Design of structures for earthquake resistance-Part 1: General rules, seismic actions and rules for buildings. *European Standard NF EN, 1:1998*.
- [3] Erlicher, S., Nguyen, Q. S. e Martin, F., 2014. Seismic design by the response spectrum method: A new interpretation of elliptical response envelopes and a novel equivalent static method based on probable linear combinations of modes. *Nuclear Engineering and Design*, **276**(May):277–294.
- [4] Fernandes, E., 1996. Computação Numérica. *Universidade do Minho*.
- [5] D'Arga e Lima, J., Monteiro, V. e Mun, M., 1985. Betão Armado - Esforços Normais e de Flexão. *Laboratório Nacional de Engenharia Civil - LNEC*.
- [6] LNEC - Laboratório Nacional de Engenharia Civil, 1983. REBAP - Regulamento de Estruturas de Betão Armado e Pré-esforçado. *RSA*.
- [7] Menun, C. e Der Kiureghian, A., 2000. Envelopes for seismic response vectors. I:theory. *Journal of Structural Engineering*, **126**(4):467–473.
- [8] Menun, C. e Der Kiureghian, A., 2000. Envelopes for seismic response vectors. II:application. *Journal of Structural Engineering*, **126**(4):474–481.
- [9] Montoya, P., Meseguer, A. e Cabré, F. M., 1981. Hormingón Armado. **11**.
- [10] Pina, H., 2010. Métodos Numéricos. *Porto Editora*.
- [11] Press, C., 1982. Manual CEB/FIP on Bending and Compression. **141**.
- [12] Ritto-Corrêa, M. e Camotim, D., 2008. On the arc-length and other quadratic control methods: Established, less known and new implementation procedures. *Computers and Structures*, **86**(11-12):1353–1368.
- [13] Ritto-Corrêa, M., Tomaz, A. R. e Guerreiro, L., 2020. The shape of the seismic response interaction diagram: The effect of combination rules in response spectrum analysis. *Earthquake Engineering and Structural Dynamics*.

- [14] Rosati, L., Marmo, F. e Serpieri, R., 2008. Enhanced solution strategies for the ultimate strength analysis of composite steel-concrete sections subject to axial force and biaxial bending. *Computer Methods in Applied Mechanics and Engineering*, **197**(9-12):1033–1055.
- [15] Sessa, S., Marmo, F. e Rosati, L., 2015. Effective use of seismic response envelopes for reinforced concrete structures. *Earthquake Engineering and Structural Dynamics*, **44**:2401–2423.
- [16] Singh, M. e Gupta, A., 1977. Design of column sections subjected to three components of earthquake. *Nuclear Engineering and Design*, **41**:129–133.
- [17] Smeby, W. e Der Kiureghian, A., 1985. Modal combination rules for multicomponent earthquake excitation. *Earthquake Engineering and Structural Dynamics*, **13**(1):1–12.

Apêndice A

Anexo1: Programa principal

```
[ ]: # -*- coding: utf-8 -*-
      """
      Created on Fri Jul 19 11:43:14 2019

      @author: AS
      """

import numpy as np
from matplotlib import pyplot as plt
from mpl_toolkits.mplot3d import axes3d
from mpl_toolkits.mplot3d.axes3d import Axes3D
from mpl_toolkits.mplot3d import proj3d
import matplotlib as mpl
import numpy as np
import matplotlib.pyplot as plt
from math import ceil
from matplotlib.path import Path
import matplotlib.patches as patches
from matplotlib import cm
class Error(Exception):
    """Base class for other exceptions"""
    pass

#-----
#-----  PROPIEDADES BETAO-----
c12={'Ecu2' : -0.0035,
      'Ec2' : -0.002,
      'Fcd' : 8000,
      'Ec' : 27000000}
c15={'Ecu2' : -0.0035,
      'Ec2' : -0.002,
      'Fcd' : 10667,
      'Ec' : 29000000}
c20={'Ecu2' : -0.0035,
      'Ec2' : -0.002,
      'Fcd' : 13333,
      'Ec' : 30000000}
c25={'Ecu2' : -0.0035,
      'Ec2' : -0.002,
      'Fcd' : 16667,
      'Ec' : 31000000}
c30={'Ecu2' : -0.0035,
      'Ec2' : -0.002,
      'Fcd' : 20000,
      'Ec' : 33000000}
c35={'Ecu2' : -0.0035,
      'Ec2' : -0.002,
      'Fcd' : 23333,
      'Ec' : 34000000}

c40={'Ecu2' : -0.0035,
      'Ec2' : -0.002,
      'Fcd' : 26667,
      'Ec' : 35000000}
c45={'Ecu2' : -0.0035,
      'Ec2' : -0.002,
      'Fcd' : 30000,
      'Ec' : 36000000}
c50={'Ecu2' : -0.0035,
```

```

        'Ec2' : -0.002,
        'Fcd' : 33333,
        'Ec' : 37000000}
concreteDetails={'C12/15' : c12,
                 'C15/20' : c15,
                 'C20/25' : c20,
                 'C25/30' : c25,
                 'C30/37' : c30,
                 'C35/45' : c35,
                 'C40/50' : c40,
                 'C45/55' : c45,
                 'C50/60' : c50}

#-----
#----- PROPRIEDADES ACO-----
a235={'Eud' : 0.01,
      'Eyd' : 0.0010217,
      'Fyd' : 204348,
      'Es' : 200000000}
a400={'Eud' : 0.01,
      'Eyd' : 0.0017391,
      'Fyd' : 347826,
      'Es' : 200000000}
a500={'Eud' : 0.01,
      'Eyd' : 0.0021739,
      'Fyd' : 434783,
      'Es' : 200000000}
steelDetails={'A235' : a235,
              'A400' : a400,
              'A500' : a500}

#-----
#-----MÉTODO DE CONVERGÊNCIA REGULA FALSI-----
def regula_falsi(funcao,a=-1,b=1,tol=1e-9,verb=False):
    fa = funcao(a)
    fb = funcao(b)
    if fa*fb>0:
        print('Intervalo inválido')
        return 0
    it = 0
    ma,mb=1,1
    while it<1000:
        it += 1
        c = (a*mb*fb-b*ma*fa)/(mb*fb-ma*fa)
        fc = funcao(c)
        ma,mb=1,1
        if abs(fc)<tol:
            break
        elif fa*fc>0:
            a,fa,mb = c,fc,1-fc/fa
            if mb<=0:
                mb=1/2
        else:
            b,fb,ma = c,fc,1-fc/fb
            if ma<=0:
                ma=1/2
    if verb:
        print(f'a={a:.4f}, b={b:.4f}, c={c:.4f}, fa={fa:.4f}, fb={fb:.4f}, fc={fc:.9f}')
    if it==1000:
        raise Error('regula falsi não convergiu')

```

```

if verb:
    print(f'it={it}, c={c:.6f}, fc={fc:.9f}')
return c
#-----
#-----PONTOS ELEMENTARES-----
class pontomaterial:
    def __init__(self,material,A,y,z):
        self.material = material
        self.A = A
        self.y = y
        self.z = z
        self.bi = np.sqrt(A)
        self.hi = np.sqrt(A)
    def deform(self,eg,chiy,chiz):
        self.e=eg+chiy*self.z-chiz*self.y
        return self.e
    def calcula_chi_max(self, eg, teta):
        chimax=1e9
        denom = (np.cos(np.radians(teta))*self.z-np.sin(np.radians(teta))*self.y)
        if abs(denom) > 1e-9:
            chimax_emax=(self.material.emax-eg)/denom
            if chimax_emax>=0:
                chimax=min(chimax,chimax_emax)
            chimax_emin=(self.material.emin-eg)/denom
            if chimax_emin>=0:
                chimax=min(chimax,chimax_emin)
        return chimax
    def calcula_lambda_max(self, a, aR, aQ):
        b=np.array([1, self.z, -self.y])
        if np.dot(b, aQ)>=0:
            deltamax = (self.material.emax-np.dot(b, a+aR))/np.dot(b, aQ)
            deltamin = (self.material.emin-np.dot(b, a+aR))/np.dot(b, aQ)
        elif np.dot(b, aQ)<0:
            deltamax = (self.material.emin-np.dot(b, a+aR))/np.dot(b, aQ)
            deltamin = (self.material.emax-np.dot(b, a+aR))/np.dot(b, aQ)
        else:
            print(aQ)
            raise('lambda_max')
        return deltamax,deltamin
    def N_min(self):
        return self.material.fmin*self.A
    def N_max(self):
        return self.material.fmax*self.A
    def forca(self,deform):
        return self.material.calcula_sigma(deform)*self.A, self.material.calcula_Ets(deform)*self.A
    def muda_centro(self,Dy,Dz):
        self.y = self.y + Dy
        self.z = self.z + Dz
    def desenha(self,ax,seccao):
        if type(self.material) in [Betao]:
            verts = [
                (self.y-self.bi/2, self.z-self.hi/2), #left, bottom
                (self.y-self.bi/2, self.z+self.hi/2), #left, top
                (self.y+self.bi/2, self.z+self.hi/2), #right, top
                (self.y+self.bi/2, self.z-self.hi/2), #right, bottom
                (self.y-self.bi/2, self.z-self.hi/2), #ignored
            ]
            codes = [

```

```

        Path.MOVETO,
        Path.LINETO,
        Path.LINETO,
        Path.LINETO,
        Path.CLOSEPOLY,
    ]
    path = Path(verts, codes)
    patch = patches.PathPatch(path, lw=0.75, facecolor='white', linestyle='--')
    ax.add_patch(patch)
    if self.y < seccao.ymin:
        seccao.ymin=self.y
    elif self.y > seccao.ymax:
        seccao.ymax=self.y
    if self.z < seccao.zmin:
        seccao.zmin=self.z
    elif self.z > seccao.zmax:
        seccao.zmax=self.z
    if type(self.material)==Betao:
        cor='orange'
        size=3
    if type(self.material)==Aco:
        cor='blue'
        size=5
    plt.plot(self.y, self.z, 'o', color=cor, markersize=size)
#-----
#-----CLASSE MATERIAL - PROPRIEDADES-----
class Material:
    def __init__(self,E):
        self.E = E
    def calcula_sigma(self,deform):
        return self.E*deform
    def mostra_grafico(self):
        pontos_traj=100
        fig, (ax1, ax2) = plt.subplots(nrows=2)
        deform_lista = []
        sigma_lista = []
        Et_lista = []
        percentage = []
        for i in range(0, pontos_traj+1):
            perc = i/pontos_traj
            percentage.append(perc*100)
            deform = self.emin*(1-perc)+self.emax*perc
            sigma = self.calcula_sigma(deform)
            Et = self.calcula_Et(deform)
            deform_lista.append(deform)
            sigma_lista.append(sigma)
            Et_lista.append(Et)
        ax1.plot(deform_lista, sigma_lista, color='red')
        ax2.plot(deform_lista, Et_lista, color='blue')
        ax1.set_xlabel('deform (%)')
        ax2.set_xlabel('deform (%)')
        ax1.set_ylabel('sigma')
        ax2.set_ylabel('Et')
        plt.tight_layout()
        plt.show()
    def calcula_E(self,deform,opcaoE='Et'):
        if opcaoE=='Et':
            Er = self.calcula_Ets(deform)

```

```

elif opcaoE=='Es':
    Er = self.calcula_Es(deform)
elif opcaoE=='E':
    Er = self.E
else:
    raise Error('Erro em calcula_E')
return Er
def calcula_Es(self,deform):
    sigma=self.calcula_sigma(deform)
    if abs(deform)<1e-6:
        secante=self.E
    else:
        secante=sigma/deform
    return secante
#-----
#-----SUBCLASSE DE MATERIAL - BETÃO-----
class Betao(Material):
    def __init__(self,C1Concrete, alfa=1.0):
        self.E = concreteDetails[C1Concrete]['Ec']
        self.alfa=alfa
        self.fcd = self.alfa*concreteDetails[C1Concrete]['Fcd']
        self.ecu2 = concreteDetails[C1Concrete]['Ecu2']
        self.ec2 = concreteDetails[C1Concrete]['Ec2']
        self.emin = 1*self.ecu2
        self.emax = -10*self.ecu2
        self.fmin = -self.fcd
        self.fmax = 0
    def calcula_sigma(self,deform):
        if deform >=0:
            sigma = 0
        elif deform >= self.ec2:
            sigma = -self.fcd*(1-(1-deform/self.ec2)**2)
        else:
            sigma = -self.fcd
        return sigma
    def calcula_Ets(self,deform):
        if deform >= -0.05*self.ec2:
            return 0
        elif deform >= 0.05*self.ec2:
            return -0.95*self.fcd/self.ec2-19*self.fcd/self.ec2**2*deform
        elif deform >= self.ec2:
            return -2*self.fcd/self.ec2*(1-deform/self.ec2)
        else:
            return 0
#-----
#-----SUBCLASSE DE MATERIAL - AÇO-----
class Aco(Material):
    def __init__(self, C1Steel):
        self.Fyd = steelDetails[C1Steel]['Fyd']
        self.E = steelDetails[C1Steel]['Es']
        self.eud = steelDetails[C1Steel]['Eud']
        self.eyd = steelDetails[C1Steel]['Eyd']
        self.emin = -1*self.eud
        self.emax = 1*self.eud
        self.fmin = -self.Fyd
        self.fmax = self.Fyd
    def calcula_sigma(self,deform):
        if abs(deform)>self.Fyd/self.E:

```

```

        return deform/abs(deform)*self.Fyd
    else:
        return self.E*deform
def calcula_Ets(self,deform):
    if deform < -self.eyd-0.05*self.eyd:
        return 0
    elif deform < -self.eyd+0.05*self.eyd:
        return 10.5*self.Fyd/self.eyd+10*self.Fyd/self.eyd**2*deform
    elif deform <= self.eyd-0.05*self.eyd:
        return self.Fyd/self.eyd
    elif deform < self.eyd+0.05*self.eyd:
        return 10.5*self.Fyd/self.eyd-10*self.Fyd/self.eyd**2*deform
    else:
        return 0

#-----
#-----CLASSE DA SECÇÃO TRANSVERSAL-----
class Seccao:
    def __init__(self,by,bz, ClConcrete, ClSteel):
        self.betao = Betao(ClConcrete)
        self.aco = Aco(ClSteel)
        self.lista_pontos = []
        self.elemRetangular = []
        self.elemCoroa = []
    def criaretangulo(self,by,bz,cy,cz,ny=0,nz=0):
        if ny==0:
            ny=ceil(by/0.1)
        if nz==0:
            nz=ceil(bz/0.1)
        for iy in range(ny):
            y = cy+(iy+1-(ny+1)/2)*by/ny
            for iz in range(nz):
                z = cz+(iz+1-(nz+1)/2)*bz/nz
                A = by*bz/(ny*nz)
                ponto=pontomaterial(self.betao,A,y,z)
                self.lista_pontos.append(ponto)
    def criaretangulo_armado(self,by,bz,cy,cz,ny=0,nz=0,c=0.05, arm_inf=[0,0], arm_sup=[0,0],
    arm_latE=[0,0], arm_latD=[0,0]):
        armadura=[arm_inf, arm_sup, arm_latE, arm_latD]
        self.criaretangulo(by,bz,cy,cz,ny,nz)
        for i, face in enumerate(armadura):
            sinal = 1
            if i%2!=0:
                sinal=-1
            if face[0] != 0:
                A = np.pi*face[1]**2/4*10**(-6)
                if i < 2:
                    z = cz + sinal*(-bz/2 + c)
                    nVar = face[0]
                    y3 = cy + by/2 - c
                    y4 = cy - by/2 + c
                    nVar-=2
                    self.lista_pontos.append(pontomaterial(self.aco, A, y3, z))
                    self.lista_pontos.append(pontomaterial(self.aco, A, y4, z))
                    if nVar != 0:
                        s = (by - 2*c)/(nVar+1)
                        y=y4
                        for i in range (0,nVar):
                            y += s

```

```

        varao = pontomaterial(self.aco, A, y, z)
        self.lista_pontos.append(varao)
    else:
        y = cy + sinal*(-by/2 + c)
        nVar = face[0]
        if nVar != 0:
            s = (bz - 2*c)/(nVar+1)
            z=cz -bz/2+c
            for i in range (0,nVar):
                z += s
                varao = pontomaterial(self.aco, A, y, z)
                self.lista_pontos.append(varao)
def criacirculo(self, d, dr=0, nd=4):
    r=d/2
    if dr==0:
        dr=0.1
    nC=ceil(r/dr)
    A = 0
    i=1
    j=0
    ri=i*r/nC
    rj=j*r/nC
    while i <= nC:
        distP=(ri+rj)/2
        nteta = ceil(2*np.pi*ri/dr/nd)*nd
        A = np.pi*(ri**2-rj**2)/nteta
        i += 1
        j += 1
        ri=i*r/nC
        rj=j*r/nC
        iteta1=1
        iteta2=0
        while iteta1 <= nteta:
            teta1 = iteta1*360/nteta
            teta2 = iteta2*360/nteta
            teta=(np.radians(teta1)+np.radians(teta2))/2
            iteta1 += 1
            iteta2 += 1
            ponto=pontomaterial(self.betao, A, distP, teta)
            self.lista_pontos.append(ponto)
def criacirculo_armado(self, d, dr=0, c=0.05, armadura=[0,0], nd=4):
    self.criacirculo(d, dr, nd)
    distP=d/2-c
    teta=0
    iteta=np.radians(360/armadura[0])
    A = np.pi*armadura[1]**2/4*10**(-6)
    if armadura[0] != 0:
        for n in range (0,armadura[0]):
            varao=pontomaterial(self.aco, A, distP, teta)
            self.lista_pontos.append(varao)
            teta += iteta
def AreaBetao(self):
    Ab=0
    for ponto in self.lista_pontos:
        if type(ponto.material) in [Betao]:
            Ab += ponto.A
    return Ab
def AreaAco(self):

```



```

As=0
for ponto in self.lista_pontos:
    if type(ponto.material) in [Aco]:
        As += ponto.A
return As
def calcula_A(self):
A=0
for ponto in self.lista_pontos:
    A += ponto.A
return A
def calcula_Sy(self):
Sy=0
for ponto in self.lista_pontos:
    Sy += ponto.A*ponto.z
return Sy
def calcula_Sz(self):
Sz=0
for ponto in self.lista_pontos:
    Sz += ponto.A*ponto.y
return Sz
def calcula_EA(self):
EA=0
for ponto in self.lista_pontos:
    EA += ponto.A*ponto.material.E
return EA
def calcula_ESy(self):
ESy=0
for ponto in self.lista_pontos:
    ESy += ponto.A*ponto.z*ponto.material.E
return ESy
def calcula_ESz(self):
ESz=0
for ponto in self.lista_pontos:
    ESz += ponto.A*ponto.y*ponto.material.E
return ESz
def calcula_Iy(self):
Iy=0
for ponto in self.lista_pontos:
    Iy += ponto.A*ponto.z**2
return Iy
def calcula_Iz(self):
Iz=0
for ponto in self.lista_pontos:
    Iz += ponto.A*ponto.y**2
return Iz
def muda_centro(self,Dy,Dz):
for ponto in self.lista_pontos:
    ponto.muda_centro(Dy,Dz)
def transf_coord(self):
for ponto in self.lista_pontos:
    (ponto.y,ponto.z)=(ponto.y*np.cos(ponto.z),ponto.y*np.sin(ponto.z))
def desenha_pontos(self):
fig, ax = plt.subplots()
self.ymin=0
self.ymax=0
self.zmin=0
self.zmax=0
for ponto in self.lista_pontos:

```

```

        ponto.desenha(ax,self)
    ax.set_xlim(self.ymin-0.1, self.ymax+0.1)
    ax.set_ylim(self.zmin-0.1, self.zmax+0.1)
    ax.set_aspect('equal', 'box')
    plt.show()
def n_varoes(self):
    n=0
    for ponto in self.lista_pontos:
        if type(ponto.material) in [Aco]:
            n += 1
    return n
def esforcos_acoEbetao(self, eg, chiy, chiz, fw):
    Na = 0
    Nb = 0
    Mya = 0
    Myb = 0
    Mza = 0
    Mzb = 0
    for ponto in self.lista_pontos:
        deform = ponto.deform(eg, chiy, chiz)
        if type(ponto.material) in [Aco]:
            forca = ponto.forca(deform)[0]*fw
            Na += forca
            Mya += forca*ponto.z
            Mza += -forca*ponto.y
        else:
            forca = ponto.forca(deform)[0]
            Nb += forca
            Myb += forca*ponto.z
            Mzb += -forca*ponto.y
    ra = np.array([Na, Mya, Mza])
    rb = np.array([Nb, Myb, Mzb])
    return ra , rb
def esforcos_acoEbetao3(self, eg, chiy, chiz, fw):
    Na = 0
    Nb = 0
    Mya = 0
    Myb = 0
    Mza = 0
    Mzb = 0
    for ponto in self.lista_pontos:
        deform = ponto.deform(eg, chiy, chiz)
        if type(ponto.material) in [Aco]:
            forca = ponto.forca(deform)[0]*fw
            Na += forca
            Mya += forca*ponto.z
            Mza += -forca*ponto.y
        else:
            forca = ponto.forca(deform)[0]
            Nb += forca
            Myb += forca*ponto.z
            Mzb += -forca*ponto.y
    N=Na+Nb
    My=Mya+Myb
    Mz=Mza+Mzb
    return N, My, Mz
def esforcos(self, eg, chiy, chiz):
    N = 0

```

```

My = 0
Mz =0
EtA = 0
for ponto in self.lista_pontos:
    deform = ponto.deform(eg,chiy,chiz)
    forca,rigidez = ponto.forca(deform)
    N += forca
    EtA += rigidez
    My += forca*ponto.z
    Mz += -forca*ponto.y
return N, My, Mz ,EtA
def calcula_KK(self, eg, chiy, chiz, opcaoE='Et',fw=1.0):
    Kxx=0
    Kxy=0
    Kxz=0
    Kyy=0
    Kyz=0
    Kzz=0
    for ponto in self.lista_pontos:
        deform = ponto.deform(eg, chiy, chiz)
        Et=ponto.material.calcula_E(deform,opcaoE)
        if type(ponto.material) in [Aco]:
            fA=fw
        else:
            fA=1
        Kxx += ponto.A*fA*Et
        Kxy += ponto.A*fA*ponto.z*Et
        Kxz += ponto.A*fA*-ponto.y*Et
        Kyy += ponto.A*fA*ponto.z**2*Et
        Kyz += ponto.A*fA*-ponto.z*ponto.y*Et
        Kzz += ponto.A*fA*ponto.y**2*Et
    KK = np.array([[Kxx, Kxy, Kxz],[Kxy, Kyy, Kyz],[Kxz, Kyz, Kzz]])
    return KK
def N_min(self):
    N = 0
    for ponto in self.lista_pontos:
        forca = ponto.N_min()
        N += forca
    return N
def N_max(self):
    N = 0
    for ponto in self.lista_pontos:
        forca = ponto.N_max()
        N += forca
    return N
def chi_maximo(self,eg,teta):
    chimax=1e9
    for ponto in self.lista_pontos:
        chimax_ponto = ponto.calcula_chi_max(eg,teta)
        chimax=min(chimax, chimax_ponto)
    return chimax
def calcula_b_comp5(self, a0):
    a0b_min,a0b_max=1e9,-1e9
    for ponto in self.lista_pontos:
        if type(ponto.material) in [Betao]:
            b=np.array([1, ponto.z, -ponto.y])
            a0b=np.dot(b, a0)
            if a0b <= a0b_min:

```

```

        b_min,a0b_min=b,a0b
        if a0b > a0b_max:
            b_max,a0b_max=b,a0b
    b_comp5=b_min+0.75*b_max
    return b_comp5
def lambda_comp5(self,a, aR, aQ, lambda0):
    lambda5max,lambda5min=1e9,-1e9
    a0 = a + aR + lambda0*aQ
    b_comp5=self.calcula_b_comp5(a0)
    if np.dot(b_comp5,aQ)>0:
        lambda5min=(self.betao.emin-np.dot(b_comp5,(a+aR)))/np.dot(b_comp5,aQ)
    if np.dot(b_comp5,aQ)<0:
        lambda5max=(self.betao.emin-np.dot(b_comp5,(a+aR)))/np.dot(b_comp5,aQ)
    return lambda5max,lambda5min
def lambda_maximo(self,a, aR, aQ):
    lambdamax,lambdamin=1e9,-1e9
    for ponto in self.lista_pontos:
        lambdamax_ponto,lambdamin_ponto = ponto.calcula_lambda_max(a, aR, aQ)
        lambdamax=min(lambdamax, lambdamax_ponto)
        lambdamin=max(lambdamin, lambdamin_ponto)
    if lambdamin>lambdamax:
        print(f'lambda_min,max={lambdamin} {lambdamax}')
    return lambdamax,lambdamin
def def_maximo(self,eg,chiy,chiz):
    defsmx=-1e9
    defcmin=1e9
    for ponto in self.lista_pontos:
        defsmx=max(defsmx, ponto.deform(eg,chiy,chiz))
        defcmin=min(defcmin, ponto.deform(eg,chiy,chiz))
    return defsmx, defcmin
def ErroN_eg(self,eg):
    chi=self.chi_maximo(eg, self.teta)
    self.chiy=chi*(np.cos(np.radians(self.teta)))
    self.chiz=chi*(np.sin(np.radians(self.teta)))
    (N1,My,Mz,EtA) = self.esforcos(eg,self.chiy,self.chiz)
    erroN=N1-self.Ntarget
    return erroN
def roda_teta(self, N, linha_N, linha_My, linha_Mz, delta_teta):
    while self.teta<360:
        N1, My, Mz = self.metodo_2(N,self.teta)
        linha_N.append(N1)
        linha_My.append(My)
        linha_Mz.append(Mz)
        self.teta += delta_teta
    return linha_N, linha_My, linha_Mz
def grafico_3D(self, pontos_N, pontos_My, pontos_Mz, N_dado='', linha_N=[], linha_My=[],
linha_Mz=[], eixo=False):
    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')
    x, y, z = pontos_My, pontos_Mz, pontos_N
    Nmax=max(pontos_N)
    Nmin=min(pontos_N)
    if eixo:
        x.append(0)
        y.append(1)
        z.append(1)
    ax.scatter(x, y, z)
    if N_dado!='':

```

```

    if N_dado >= Nmin and N_dado <= Nmax:
        x, y, z = linha_My, linha_Mz, linha_N
        x.append(linha_My[0])
        y.append(linha_Mz[0])
        z.append(linha_N[0])
        plt.plot(x, y, z, ls='-', c='r')
        fig2=plt.figure()
        plt.title(f'Curva de interação My - Mz para N={N_dado:.2f}kN')
        plt.plot(x, y, ls='-', lw='1', marker='o', ms=7, fillstyle='full')
        plt.plot(250, 150, lw='3', marker='o', c='r')
        plt.xlabel('My [kNm]')
        plt.ylabel('Mz [kNm]')
    plt.show()
    def grafico_3D_3listas(self, listaN1, listaMy1, listaMz1, listaN2, listaMy2, listaMz2,
→listaN3, listaMy3, listaMz3):
        fig = plt.figure()
        ax = fig.add_subplot(111, projection='3d')
        ax.scatter(listaMy1, listaMz1, listaN1, c='r')
        ax.scatter(listaMy2, listaMz2, listaN2, c='g')
        ax.scatter(listaMy3, listaMz3, listaN3, c='b')
        plt.show()
#-----
#-----MÉTODO 1-----
def aplica_chi_teta_N(self, chi_f, teta, N, pontos_traj = 100):
    fig, (ax1, ax2, ax3) = plt.subplots(nrows=3)
    N1_lista = []
    My_lista = []
    Mz_lista = []
    eg_lista = []
    erro_lista = []
    percentage = []
    it_tot = 0
    for i in range(0, pontos_traj+1):
        perc = i/pontos_traj
        chi = chi_f*perc #para obter trajetória chi - momento
        chiy=chi*np.cos(np.radians(teta))
        chiz=chi*np.sin(np.radians(teta))
        eg=0 #condição inicial
        it = 0
        while True:
            it += 1
            it_tot +=1
            (N1,My,Mz,EtA) = self.esforcos(eg,chiy,chiz)
            (esmax,ecmin) = self.def_maximo(eg,chiy,chiz)
            erroN=N1-N
            if abs(erroN)<1e-6:
                break
            eg = eg - erroN/EtA
        if esmax>self.aco.eud or ecmin<self.betao.ecu2:
            break
        percentage.append(perc*100)
        N1_lista.append(N1)
        My_lista.append(My)
        Mz_lista.append(Mz)
        eg_lista.append(eg*1000)
        erro_lista.append(abs(erroN))
    ax1.plot(percentage, My_lista, color='red')
    ax2.plot(percentage, Mz_lista, color='blue')

```

```

ax3.plot(percentage, eg_lista, color='green')
ax1.set_xlabel('Percentage (%)')
ax2.set_xlabel('Percentage (%)')
ax3.set_xlabel('Percentage (%)')
ax1.set_ylabel('My [kNm]')
ax2.set_ylabel('Mz [kNm]')
ax3.set_ylabel('eg (%)')
plt.tight_layout()
plt.show()
print(f'it={it} it_tot={it_tot} eg={eg:.4f} N1={N1:.2f} My={My:.2f} Mz={Mz:.2f}')
return My

#-----MÉTODO 2-----
def metodo_2(self, N, teta):
    Neg = lambda eg: self.ErroN_eg(eg)
    self.Ntarget=N
    self.teta=teta
    eg = regula_falsi(Neg,-0.010,0.010, verb=True)
    (N1,My,Mz,EtA) = self.esforcos(eg, self.chiy, self.chiz)
    return N1, My, Mz

#-----MÉTODO 3-----
def find_lambdda_Advanced(self, N1=0, My1=0, Mz1=0, N0=0, My0=0, Mz0=0, fw=1, verbose=False):
    #Condições iniciais a=(deform, chiy, chiz)
    a=np.array([0.0, 0.0, 0.0])
    esforcos_dados_0=np.array([N0, My0, Mz0])
    esforcos_dados_1=np.array([N1, My1, Mz1])
    Q=np.array([N1, My1, Mz1])
    fator=1
    it=0
    itmax=20000
    while True and (it<itmax):
        it +=1
        s = esforcos_dados_0 + esforcos_dados_1*fator
        #Funções lineares
        eg,chiy,chiz = a[0], a[1], a[2]
        ra, rb = self.esforcos_acoEbetao(eg, chiy, chiz, 1)
        r = fw*ra + rb
        #Vetor residuo
        R_a = r-s
        erroR = np.linalg.norm(R_a)
        if erroR<1e-9:
            if verbose:
                print(f'it={it}')
                print(f'fator={fator}')
                print(f'eg,chiy,chiz={eg,chiy,chiz}')
                print(f'R={R_a}')
                print(f'erroR={erroR}')
            break
        if it==itmax:
            raise Error('find lambda não converge')
    #Matriz rigidez
    KK = self.calcula_KK(eg, chiy, chiz, opcaoE='Et', fw=fw)
    if it==1:
        KKO = 1.0*KK
        det0 = np.linalg.det(KKO)
    #Matriz dR/da
    detsdet0 = np.linalg.det(KK)/det0

```

```

if detsdet0>0.1:
    dR_da = KK
else:
    KKs = self.calcula_KK(eg, chiy, chiz, opcaoE='Es', fw=fw)
    dR_da = KKs
    if verbose:
        print(f'vou usar a secante detsdet0={detsdet0:.6f}')
#Inversa da matriz dR_da
    dR_da_I = np.linalg.inv(dR_da)
#Incremento de a
    aR = -dR_da_I@R_a
    aQ = dR_da_I@Q
    delta_fator0, fmin=self.lambda_maximo(a, aR, aQ)
    delta_fator5, fmin=self.lambda_comp5(a, aR, aQ, delta_fator0)
    if verbose:
        print(f'delta_fator0={delta_fator0:.4f} delta_fator5={delta_fator5:.4f}')
    delta_fator=min(delta_fator0,delta_fator5)
#Vetor solução
    a += aR + delta_fator*aQ
    fator += delta_fator
def_max,def_min= self.def_maximo(eg,chiy,chiz)
    if verbose:
        print(f'def_max={def_max:.6f}, def_min={def_min:.6f}')
    return fator, it
#-----
#-----MÉTODO 4-----
def find_w(self, N=0, My=0, Mz=0, verbose=False):
    fw=1
    KKO = self.calcula_KK(0, 0, 0) #matriz elastica
    det0 = np.linalg.det(KKO)
    KKO_I = np.linalg.inv(KKO)
    s = np.array([N, My, Mz])
    a = KKO_I@s
    it=0
    itmax=5000
    while True and (it<itmax):
        it +=1
        #Funções lineares
        eg,chiy,chiz = a[0], a[1], a[2]
        ra, rb = self.esforcos_acoEbetao(eg,chiy,chiz,1)
        r = fw*ra + rb
        #Vetor residuo
        R_a = r-s
        erroR = np.linalg.norm(R_a)
        if erroR<1e-8:
            return fw, it
            break
        if it==itmax:
            print(f'Erro={erroR}')
            raise Error('find w não converge')
    Q = -ra
    #Matriz rigidez
    KK = self.calcula_KK(eg, chiy, chiz, opcaoE='Et',fw=fw)
    detsdet0 = np.linalg.det(KK)/det0
    #Matriz dR/da
    if detsdet0>0.1:
        dR_da = KK
    else:

```

```

        KKs = self.calcula_KK(eg, chiy, chiz, opcaoE='Es')
        detssdet0 = np.linalg.det(KKs)/det0
        if detssdet0>0.001:
            dR_da = KKs
        else:
            if verbose:
                print(f'vou usar a elástica detssdet0={detssdet0:.6f}')
            dR_da = KK0
        #Inversa da matriz dR_da
        dR_da_I = np.linalg.inv(dR_da)
        #Incremento de a
        aR = -dR_da_I@R_a
        aQ = dR_da_I@Q
        naR = np.linalg.norm(aR)
        na = np.linalg.norm(a)
        naRna_lim=0.5*5
        if naR/na>naRna_lim:
            if verbose:
                print(f'naR/na={naR/na:.2f} -> {naRna_lim:.2f}')
            aR = naRna_lim*na/naR*aR
        fmax, delta_fw0 = self.lambda_maximo(a, aR, aQ)
        fmax, delta_fw5=self.lambda_comp5(a, aR, aQ, delta_fw0)
        delta_fw=max(delta_fw0,delta_fw5)
        #Vetor solução
        a += aR + delta_fw*aQ
        fw += delta_fw
    def_max,def_min= self.def_maximo(eg,chiy,chiz)
    if verbose:
        print(f'def_max={def_max:.6f}, def_min={def_min:.6f}')
    return fw, it
#-----
#-----MÉTODO 5-----
def met5_lambdda(self, NsM=1):
    #Define uma accao em particular
    s0,S =accao2()
    s0,S=limpa_Mz(s0,S)
    #construção do elipsoide
    lista_N=[]
    lista_My=[]
    lista_N2=[]
    lista_My2=[]
    lista_N3=[]
    lista_My3=[]
    lambddaMax=100
    for teta in np.arange(0,361,1.5):
        t = np.array([np.cos(np.radians(teta)),np.sin(np.radians(teta)),0])
        s = calcula_sv(s0,S,t)
        N0,My0=s0[0],s0[1]
        N,My=s[0],s[1]
        s1 = calcula_s1v(S,t)
        N1,My1=s1[0],s1[1]
        lambdda, it=secr.find_lambdda_Advanced(N1=N1, My1=My1, NO=N0, My0=My0)
        lambddaMax=min(lambddaMax, lambdda)
        lista_N.append(N)
        lista_My.append(My)
        s2=calcula_sv(s0,S,t,lamb=lambdda)
        N2,My2=s2[0],s2[1]
        lista_N2.append(N2)

```



```

        lista_My2.append(My2)
for teta in np.arange(0,361,3):
    t = np.array([np.cos(np.radians(teta)),np.sin(np.radians(teta)),0])
    s3=calcula_sv(s0,S,t,lamb=lambddaMax)
    N3,My3=s3[0],s3[1]
    lista_N3.append(N3)
    lista_My3.append(My3)
fig = plt.figure()
plt.plot(lista_My,lista_N)
fig2 = plt.figure()
plt.plot(lista_My,lista_N)
plt.plot(lista_My2, lista_N2, c='r')
fig3 = plt.figure()
plt.plot(lista_My,lista_N)
plt.plot(lista_My2, lista_N2, c='r')
plt.plot(lista_My3, lista_N3, c='g')
plt.show()
print(f'Lambda máximo={lambddaMax}')
def met5_lambdda_3D(self, NsM=1):
    #Define uma accao em particular
    s0,S =accas2()
    #construção do elipsoide
    lista_N=[]
    lista_My=[]
    lista_Mz=[]
    lista_N2=[]
    lista_My2=[]
    lista_Mz2=[]
    lista_N3=[]
    lista_My3=[]
    lista_Mz3=[]
    lambddaMax=100
    t3=0
    while t3<=180:
        teta=0
        for teta in np.arange(0,360,10):
            t = np.array([np.cos(np.radians(t3))*NsM,
                          np.cos(np.radians(teta))*np.sin(np.radians(t3)),
                          np.sin(np.radians(teta))*np.sin(np.radians(t3))])
            s = calcula_sv(s0,S,t)
            N0,My0, Mz0=s0[0],s0[1],s0[2]
            N,My, Mz=s[0],s[1],s[2]
            s1 = calcula_s1v(S,t)
            N1,My1,Mz1=s1[0],s1[1],s1[2]
            lambdda, it=secre.find_lambdda_Advanced(N1=N1, My1=My1, Mz1=Mz1, N0=N0, My0=My0,
→Mz0=Mz0)

            print(f's={s[0]:6.2f}; {s[1]:6.2f}; {s[2]:6.2f}; lambdda={lambdda:6.2f}, it={it}')
            lambddaMax=min(lambddaMax, lambdda)
            lista_N.append(N)
            lista_My.append(My)
            lista_Mz.append(Mz)
            s2=calcula_sv(s0,S,t,lamb=lambdda)
            N2,My2,Mz2=s2[0],s2[1],s2[2]
            lista_N2.append(N2)
            lista_My2.append(My2)
            lista_Mz2.append(Mz2)
            print(teta)
        t3+=2

```

```

        print(t3)
t3=0
while t3<=180:
    teta=0
    for teta in np.arange(0,360,10):
        t = np.array([np.cos(np.radians(t3))*NsM,
                      np.cos(np.radians(teta))*np.sin(np.radians(t3)),
                      np.sin(np.radians(teta))*np.sin(np.radians(t3))])
        s3=calcula_sv(s0,S,t,lamb=lambddaMax)
        N3,My3,Mz3=s3[0],s3[1],s3[2]
        lista_N3.append(N3)
        lista_My3.append(My3)
        lista_Mz3.append(Mz3)
        print(teta)
    t3+=2
    print(t3)
print(f'Lambda máximo={lambddaMax}')
self.grafico_3D_3listas(lista_N, lista_My, lista_Mz, lista_N2, lista_My2, lista_Mz2,
→lista_N3, lista_My3, lista_Mz3)
#-----
#-----MÉTODO 6 -----
def met6_w(self, NsM=1):
    #Define uma accao em particular
    s0,S =accas2()
    s0,S=limpa_Mz(s0,S)
    #construção do elipsoide
    lista_N=[]
    lista_My=[]
    lista_N2=[]
    lista_My2=[]
    lista_N3=[]
    lista_My3=[]
    fwMin=1E-9
    for teta in np.arange(0,361,5):
        t = np.array([np.cos(np.radians(teta))*NsM,np.sin(np.radians(teta)),0])
        s = calcula_sv(s0,S,t)
        N,My=s[0],s[1]
        fw, it=secr.find_w(N=N, My=My)
        fwMin=max(fwMin, fw)
        lista_N.append(N)
        lista_My.append(My)
    for teta in np.arange(0,361,5):
        t = np.array([np.cos(np.radians(teta))*NsM,np.sin(np.radians(teta)),0])
        N1,My1=t[0],t[1]
        lambdda, it=secr.find_lambdda_Advanced(N1=N1, My1=My1, fw=1)
        lista_N2.append(N1*lambdda)
        lista_My2.append(My1*lambdda)
        lambdda, it=secr.find_lambdda_Advanced(N1=N1, My1=My1, fw=fwMin)
        lista_N3.append(N1*lambdda)
        lista_My3.append(My1*lambdda)
    fig = plt.figure()
    plt.plot(lista_My, lista_N)
    plt.plot(lista_My2, lista_N2, c='r')
    plt.plot(lista_My3, lista_N3, c='g')
    plt.show()
def met6_w_3D(self, NsM=1):
    #Define uma accao em particular
    s0,S =accas2()

```

```

#construção do elipsoide
lista_N=[]
lista_My=[]
lista_Mz=[]
lista_N2=[]
lista_My2=[]
lista_Mz2=[]
lista_N3=[]
lista_My3=[]
lista_Mz3=[]
fwMin=1E-9
#Lista de pontos do elipsóide e find_fw_min
t3=0
while t3<=180:
    teta=0
    for teta in np.arange(0,360,15):
        t = np.array([np.cos(np.radians(t3))*NsM,
                    np.cos(np.radians(teta))*np.sin(np.radians(t3)),
                    np.sin(np.radians(teta))*np.sin(np.radians(t3))])
        s = calcula_sv(s0,S,t) #s(v) em vez de s(t)
        print(f't={t[0]:6.2f}; {t[1]:6.2f}; {t[2]:6.2f}')
        print(f's={s[0]:6.2f}; {s[1]:6.2f}; {s[2]:6.2f}')
        N,My,Mz=s[0], s[1], s[2]
        fw, it=seccr.find_w(N=N, My=My, Mz=Mz)
        print(f's={s[0]:6.2f}; {s[1]:6.2f}; {s[2]:6.2f}; fw={fw:6.2f}, it={it}')
        fwMin=max(fwMin, fw)
        lista_N.append(N)
        lista_My.append(My)
        lista_Mz.append(Mz)
    t3+=5
#ciclo para desenhar as superficies de interação
t3=0
while t3<=180:
    teta=0
    for teta in np.arange(0,360,15):
        t = np.array([np.cos(np.radians(t3))*NsM,
                    np.cos(np.radians(teta))*np.sin(np.radians(t3)),
                    np.sin(np.radians(teta))*np.sin(np.radians(t3))])
        N1,My1,Mz1=t[0],t[1],t[2]
        lambdda, it=seccr.find_lambdda_Advanced(N1=N1, My1=My1, Mz1=Mz1, fw=1)
        lista_N2.append(N1*lambdda)
        lista_My2.append(My1*lambdda)
        lista_Mz2.append(Mz1*lambdda)
        lambdda, it=seccr.find_lambdda_Advanced(N1=N1, My1=My1, Mz1=Mz1, fw=fwMin)
        lista_N3.append(N1*lambdda)
        lista_My3.append(My1*lambdda)
        lista_Mz3.append(Mz1*lambdda)
    t3+=5
self.grafico_3D_3listas(lista_N, lista_My, lista_Mz, lista_N2, lista_My2, lista_Mz2,
→lista_N3, lista_My3, lista_Mz3)

print(f'fw_min={fwMin}')
#-----
#-----Curvas de interação-----
def curvas(self, lista_taxas=[1.0], NsM=1):
    listaN=[]
    listaMy=[]
    listaMz=[]

```

```

for arma in lista_taxas:
    Aa=self.AreaAco()
    Ab=self.AreaBetao()
    fw=arma/self.aco.Fyd*self.betao.fcd/Aa*Ab
    i=0
    while i<=180:
        j=0
        while j<360:
            x=np.cos(np.radians(j))*np.sin(np.radians(i))
            y=np.sin(np.radians(j))*np.sin(np.radians(i))
            z=np.cos(np.radians(i))*NsM
            fac, it=self.find_lambdda_Advanced(N1=z, My1=x, Mz1=y, fw=fw, verbose=False)
            print()
            print(f'a={arma:.2f} i={i} j={j} x={x:.3f}, y={y:.3f}, z={z:.3f}, it={it}')
            N, My, Mz=fac*z, fac*x, fac*y
            print(f'lambda={fac:6.2f} fw={fw:6.2f} N={N:8.2f}, My={My:8.2f}, Mz={Mz:8.2f}')
            listaN.append(N)
            listaMy.append(My)
            listaMz.append(Mz)
            j+=15
        i+=5
    return listaN, listaMy, listaMz
def curvas_3D_desenho(self, lista_taxas=[1.0], NsM=10):
    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')
    c=['r', 'g', 'b', 'k']
    y=0
    for t in lista_taxas:
        listaN, listaMy, listaMz = self.curvas(lista_taxas=[t], NsM=NsM)
        ax.scatter(listaMy, listaMz, listaN, c=c[y])
        y+=1
    ax.set_xlabel('My [kNm]')
    ax.set_ylabel('Mz [kNm]')
    ax.set_zlabel('N [kN]')
    plt.show()
def curvas_2D(self, lista_taxas=[1.0], NsM=1):
    listaN=[]
    listaMy=[]
    for arma in lista_taxas:
        Aa=self.AreaAco()
        Ab=self.AreaBetao()
        fw=arma/self.aco.Fyd*self.betao.fcd/Aa*Ab
        print(f'arma={arma:8.4f} fw={fw:8.2f}')
        j=0
        while j<361:
            x=np.cos(np.radians(j))
            y=np.sin(np.radians(j))
            fac, it=self.find_lambdda_Advanced(N1=y, My1=x, fw=fw, verbose=False)
            print()
            print(f'a={arma:.2f} j={j} x={x:.3f}, y={y:.3f}, it={it}')
            N, My=fac*y, fac*x
            print(f'lambda={fac:6.2f} fw={fw:6.2f} N={N:8.2f} My={My:8.2f}')
            listaN.append(N)
            listaMy.append(My)
            j+=2
    return listaN, listaMy
def curvas_2D_desenho(self, taxaI, taxaF, intervalo):
    fig, ax=plt.subplots()

```

```

t=taxaI
r=0
while t <= taxaF:
    cor='b'
    label=''
    if (ceil(t*10))%5==0:
        r-=100
        label=str(round(t,1))
        cor='r'
    l1, l2 = self.curvas_2D(lista_taxas=[t])
    ax.plot(l2, l1, c=cor, label=label)
    ax.text(l2[0], l1[0]-300+r, label)
    t+=intervalo
ax.axhline(y=0, color='k')
ax.axvline(x=0, color='k')
ax.grid(True, which='both')
plt.xlabel('N [kNm]')
plt.ylabel('My [kNm]')
plt.show()

#-----
#-----SUBCLASSE DA SECÇÃO-RETANGULAR-----

class SeccaoR(Seccao):
    def __init__(self,by,bz, betao, aco ,ny=0,nz=0, c=0.05, arm_inf=[0,0], arm_sup=[0,0],u
    →arm_latE=[0,0], arm_latD=[0,0]):
        self.betao = betao
        self.aco = aco
        self.lista_pontos = []
        self.criaretangulo_armado(by,bz,0,0,ny,nz,c, arm_inf, arm_sup, arm_latE, arm_latD)
        zg= self.calcula_ESy()/self.calcula_EA()
        yg= self.calcula_ESz()/self.calcula_EA()
        self.muda_centro(-yg,-zg)
        self.desenha_pontos()

#-----
#-----SUBCLASSE DA SECÇÃO-T-----

class SeccaoT(Seccao):
    def __init__(self,bf,hw,tw,tf, betao, aco, ny=0,nz=0, c=0.05, arm_inf1=[0,0], arm_sup1=[0,0],u
    →arm_latE1=[0,0], arm_latD1=[0,0], arm_inf2=[0,0], arm_sup2=[0,0], arm_latE2=[0,0],u
    →arm_latD2=[0,0]):
        self.betao = betao
        self.aco = aco
        self.lista_pontos = []
        self.criaretangulo_armado(bf,tf,0,hw+tf/2,ny,nz,c,arm_inf1,arm_sup1,arm_latE1,arm_latD1)
        self.criaretangulo_armado(tw,hw,0,hw/2,ny,nz,c,arm_inf2,arm_sup2,arm_latE2,arm_latD2)
        zg= self.calcula_ESy()/self.calcula_EA()
        yg= self.calcula_ESz()/self.calcula_EA()
        self.muda_centro(-yg,-zg)
        self.desenha_pontos()

#-----
#-----SUBCLASSE DA SECÇÃO-CIRCULAR-----

class SeccaoC(Seccao):
    def __init__(self, d, betao, aco,dr=0, c=0.05, armadura=[0,0], nd=4):
        self.betao = betao
        self.aco = aco
        self.lista_pontos = []
        self.criacirculo_armado(d,dr,c, armadura, nd)
        self.transf_coord()

```

```

        self.desenha_pontos()
##-----
#-----SUPERFICIES DE INTERAÇÃO-----
def elipsoide(NsM=1):
    #Define uma accao em particular
    s0,S =accao2()
    #construção do elipsoide
    lista_N=[]
    lista_My=[]
    lista_Mz=[]
    t3=0
    while t3<=180:
        teta=0
        for teta in np.arange(0,360,5):
            t = np.array([np.cos(np.radians(t3))*NsM,
                          np.cos(np.radians(teta))*np.sin(np.radians(t3)),
                          np.sin(np.radians(teta))*np.sin(np.radians(t3))])
            s = calcula_sv(s0,S,t)
            print(f't={t[0]:6.2f}; {t[1]:6.2f}; {t[2]:6.2f}')
            print(f's={s[0]:6.2f}; {s[1]:6.2f}; {s[2]:6.2f}')
            N,My,Mz=s[0], s[1], s[2]
            lista_N.append(N)
            lista_My.append(My)
            lista_Mz.append(Mz)
        t3+=5
    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')
    ax.scatter(lista_My, lista_Mz, lista_N, c='r')
    plt.show()
#-----
#-----ROTINAS PARA AÇÕES SÍSMICAS-----
def accao2():
    s0 = np.array([-0,-100,-100])
    freq = np.array([1.25, 1.27, 1.64, 3.70, 3.76, 4.81, 5.94, 6.00])
    ro2 = np.zeros((8,8)) #coeficientes CQC reais
    for i,fi in enumerate(freq):
        for j,fj in enumerate(freq):
            r = fi/fj
            c2 = 0.05**2
            ro2[i,j] = (8*c2*(1+r)*r**1.5)/((1-r)**2+4*c2*r*(1+r)**2)
    #sismox
    sx1 = np.array([250,100,-5])
    sx2 = np.array([0,0,0])
    sx3 = np.array([8,8,4])
    sx4 = np.array([-18,16,-1])
    sx5 = np.array([0,0,0])
    sx6 = np.array([-1,1,1])
    sx7 = np.array([1,4,-1])
    sx8 = np.array([0,0,0])
    #sismoxy
    sy1 = np.array([0,0,0])
    sy2 = np.array([272,0,-118])
    sy3 = np.array([0,0,0])
    sy4 = np.array([0,0,0])
    sy5 = np.array([-19,0,-17])
    sy6 = np.array([0,0,0])
    sy7 = np.array([0,0,0])
    sy8 = np.array([1,0,-4])

```

```

lista_sx=[sx1,sx2,sx3,sx4,sx5,sx6,sx7,sx8]
lista_sy=[sy1,sy2,sy3,sy4,sy5,sy6,sy7,sy8]
lista_sg=[lista_sx,lista_sy]
S2 = np.zeros((3,3))
for sk in lista_sg:
    for i,ski in enumerate(sk):
        for j,skj in enumerate(sk):
            S2 += ro2[i,j]*np.outer(ski,skj)
return s0,S2
def limpa_Mz(s0,S):
s0[2]=0
S[0,2],S[1,2],S[2,1],S[2,0]=0,0,0,0
return s0,S
def accaol():
s0 = np.array([-150,50,0])
#sismoæ
sx1 = np.array([-800,30,0])
sx2 = np.array([0,150,0])
sx3 = np.array([10,0,50])
lista_sx=[sx1,sx2,sx3]
lista_sg=[lista_sx]
#versão SRSS
S = np.zeros((3,3))
for sk in lista_sg:
    for ski in sk:
        S += np.outer(ski,ski)
return s0,S
def calcula_s(s0,S,t,lamb=1):
St = S@t
denom = np.sqrt(np.dot(St,t))
s=s0+(lamb/denom)*St
return s
def calcula_s1(S,t):
St = S@t
denom = np.sqrt(np.dot(St,t))
s1=(1/denom)*St
return s1
def calcula_sv(s0,S,v,lamb=1):
Sinv = np.linalg.inv(S)
Sinv_v = Sinv@v
denom = np.sqrt(np.dot(Sinv_v,v))
s=s0+(lamb/denom)*v
return s
def calcula_s1v(S,v):
Sinv = np.linalg.inv(S)
Sinv_v = Sinv@v
denom = np.sqrt(np.dot(Sinv_v,v))
s=(1/denom)*v
return s
def st_sv(NsM=1):
s0,S =accaol()
s0,S=limpa_Mz(s0,S)
lista_Nt=[]
lista_Myt=[]
lista_Nv=[]
lista_Myv=[]
for teta in np.arange(0,360,1):
t = np.array([np.cos(np.radians(teta))*NsM,np.sin(np.radians(teta)),0])

```

```
v = np.array([np.cos(np.radians(teta))*NsM,np.sin(np.radians(teta)),0])
st = calcula_s(s0,S,t)
sv = calcula_sv(s0,S,v)
Nt,Myt=st[0],st[1]
Nv,Myv=sv[0],sv[1]
lista_Nt.append(Nt)
lista_Myt.append(Myt)
lista_Nv.append(Nv)
lista_Myv.append(Myv)
fig = plt.figure()
plt.plot(lista_Myt, lista_Nt, c='b')
plt.plot(lista_Myv, lista_Nv, c='g')
plt.show()
```