



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências

Faculdade de Engenharia

Hugo Lobato Melo


Sistema de automatização para ensino de Elementos de Máquinas

Rio de Janeiro

2022

Hugo Lobato Melo

Sistema de automatização para ensino de Elementos de Máquinas



Projeto de graduação apresentado,
como requisito parcial para obtenção
do grau de Engenheiro Mecânico, à
Faculdade de Engenharia, da
Universidade do Estado do Rio de
Janeiro.

Orientador: Prof. Carlos Cezar de La Plata Ruiz

Rio de Janeiro

2022

CATALOGAÇÃO NA FONTE
UERJ / REDE SIRIUS / BIBLIOTECA CTC/B

Ficha elaborada pelo autor através do
Sistema para Geração Automática de Ficha Catalográfica da Rede Sirius - UERJ

M528 Melo, Hugo Lobato.
 Sistema de automatização para ensino de Elementos
 de Máquinas / Hugo Lobato Melo. - 2022.
 77 f.

 Orientador: Carlos Cezar de La Plata Ruiz.
 Trabalho de Conclusão de Curso apresentado à
Universidade do Estado do Rio de Janeiro, Faculdade
de Engenharia, para obtenção do grau de bacharel em
Engenharia Mecânica.

 1. Automatização - Monografias. 2. Elementos de
 máquinas - Monografias. 3. Engenharia mecânica -
 Monografias. I. Ruiz, Carlos Cezar de La Plata. II.
 Universidade do Estado do Rio de Janeiro. Faculdade
 de Engenharia. III. Título.

CDU 621

Hugo Lobato Melo

Sistema de automatização para ensino de Elementos de Máquinas

Projeto de graduação apresentado,
como requisito parcial para obtenção
do grau de Engenheiro Mecânico, à
Faculdade de Engenharia, da
Universidade do Estado do Rio de
Janeiro.

Aprovado em: 15 de Dezembro de 2022

Banca Examinadora:



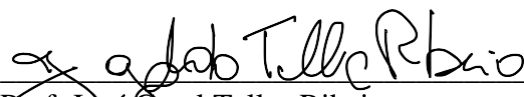
Prof. Carlos Cezar de La Plata Ruiz

Faculdade de Engenharia - UERJ



Prof. Francisco José da Cunha Pires Soeiro

Faculdade de Engenharia - UERJ



Prof. José Geral Telles Ribeiro

Faculdade de Engenharia – UERJ



Prof. Flavio Dupim Pires

Transpetro

Rio de Janeiro

2022

AGRADECIMENTOS

Agradeço, primeiramente, à minha namorada e futura esposa, Natalye Bacanieski, que apoia e incentiva minhas ideias e decisões de vida e, durante o desenvolvimento deste trabalho, seguiu sendo minha companheira, auxiliando com ideias e motivação.

Ao meu professor orientador, La Plata, pela paciência e dedicação do seu tempo em reuniões de orientação, em que sempre buscou transmitir seu conhecimento da melhor forma e com toda calma necessária, tornando-se parte ativa e fundamental para a conclusão do trabalho.

Agradeço, também, à minha família que me deu suporte para que eu chegasse ao final da graduação e a todos os meus amigos que compartilharam momentos diversos ao longo da minha graduação, desde estudos até as festas, tornando tudo mais leve.

RESUMO

Melo, Hugo. *Sistema de automatização para ensino de Elementos de Máquinas*. 2022 77f. Trabalho de Conclusão de Curso (Graduação em Engenharia Mecânica) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2022.

Neste projeto final foi feito a automatização dos exercícios dados em sala pelo professor utilizando códigos em Python, uma das linguagens de programação mais usadas por desenvolvedores quando é necessário automatizar algum processo. Sua idealização surgiu devido à pandemia causada pelo vírus SARS-Cov-2, época em que ensino EAD foi aplicado na Universidade. Junto a isso, diversas dificuldades surgiram, já que a convivência presencial entre professor e aluno não era possível e, conseqüentemente, as dúvidas não eram facilmente sanadas. O objetivo do trabalho, além da automação, foi auxiliar os alunos no processo de aprendizagem utilizando uma nova estratégia a fim de analisar/criticar o resultado do exercício já que em alguns casos as respostas não são encontradas na apostila usada pelo professor. No desenvolvimento do trabalho, primeiro é feito um resumo teórico para explicar o motivo das equações serem utilizadas e em seguida o código é apresentado.

Palavras-chave: Python, Automatização, Elementos de máquinas.

ABSTRACT

Melo, Hugo. *Automation system for teaching Machine Elements*. 2022 77f. Trabalho de Conclusão de Curso (Graduação em Engenharia Mecânica) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2022.

In this final project, the exercises given in the classroom by the teacher were automated using codes in Python, one of the most used programming languages by developers when it is necessary to automate some process. Its idealization arose due to the pandemic caused by COVID-19, when distance learning was applied at the University. Along with this, several difficulties arose, since face-to-face interaction between teacher and student was not possible and, consequently, doubts were not easily resolved. The objective of the work, in addition to automation, was to assist students in the learning process using a new strategy in order to analyze/criticize the result of the exercise since in some cases the answers are not found in the handout used by the teacher. In the development of the work, first a theoretical summary is made to explain why the equations are used and then the code is presented.

Keywords: Python, Automation, Machine Elements.

LISTA DE FIGURAS

Figura 1: Estrutura condicional	10
Figura 2: Exemplo de laço de repetição	11
Figura 3: Paralelepípedo elementar (BUDYNAS, 2011)	13
Figura 4: Estado plano de tensão (BUDYNAS, 2011)	13
Figura 5: Novos componentes de tensão quando o sistema de referência é girado de um ângulo ϕ (BUDYNAS, 2011).....	14
Figura 6: Corpo cilíndrico sujeito a esforços combinados.	17
Figura 7: Elemento submetido a uma carga axial F	17
Figura 8: Elemento submetido a um momento de torção T puro.....	18
Figura 9: Elemento submetido a um momento de flexão M puro.....	19
Figura 10: Tensão plana de um eixo sujeito a esforços combinados (LA PLATA, 2022)	20
Figura 11: Representação da estrutura de L invertido com suas dimensões e uma carga axial F aplicada (LA PLATA, 2022).....	21
Figura 12: Máquina de flexões alternadas utilizadas no ensaio de fadiga (GARCIA, 2012)	23
Figura 13: Dados do ensaio computados em um gráfico (BUDYNAS, 2011)	24
Figura 14: Relações de tempo-tensão. (d) Tensão variada (e) Tensão repetida (f) Tensão alternada (BUDYNAS, 2011).....	29
Figura 15: Diagrama de Sorderberg (LA PLATA, 2022).....	31
Figura 16: Critério Modificado de Goodman (LA PLATA, 2022)	32
Figura 17: Critério de Gerber (arco de parábola) (LA PLATA, 2022).....	33
Figura 18: Critério ASME (LA PLATA, 2022)	33
Figura 19: Critério do Escoamento Ampliado (LA PLATA, 2022)	34
Figura 20: Ilustração da figura utilizada no exercício de projeto do capítulo 5. Projeto de uma peça sujeita a um esforço variável (LA PLATA, 2022).....	35
Figura 21: Medidas principais da rosca de um parafuso (SHIGLEY, 1984)	44
Figura 22: Montagem com pré-carga (SHIGLEY, 1984).....	45
Figura 23: Esquemática de sistemas mecanicamente equivalentes	50
Figura 24: Direção das cargas P_i e Q_i	50
Figura 25: Representação da montagem excêntrica utilizada no exercício automatizado desta seção	51
Figura 26: Ilustração de uma árvore principal (REAL TRATOR PEÇAS, 2022).....	59
Figura 27: Esforços combinados em uma árvore (LA PLATA, 2022)	59
Figura 28: Ilustração do diâmetro interno (d_i) e externo (d_o) de uma árvore (ANDRADE, 2022).....	60
Figura 29: Demonstração de um eixo de comprimento L , sofrendo uma deflexão angular (θ) causada por um carregamento P . O primeiro está sem deformação. O segundo está sofrendo uma deflexão devido a força aplicada.....	62
Figura 30: Representação da chaveta em um sistema com eixo e um outro componente (Chaveta mecânica: veja para que serve, principais tipos e aplicações., 2022)	65
Figura 31: Ligação estriada entre eixo e bucha (Eixos e buchas estriadas, 2022)	65
Figura 32: Representação esquemática de uma estria com uma força F aplicada (LA PLATA, 2022)	66

Figura 33: Mancal de deslizamento (BUDYNAS, 2011)	68
Figura 34: Gráfico de k em função da razão L/D (HALL, 1981)	69
Figura 35: Ilustração do diâmetro e comprimento do mancal de deslizamento (LA PLATA, 2022)	70
Figura 36: Carga radial aplicada sobre o mancal. Figura adaptada de (ANDRADE, 2022)	70
Figura 37: Mancal de rolamento (BUDYNAS, 2011)	73

LISTA DE TABELAS

Tabela 1: Valores de a e b para determinação do ka (BUDYNAS, 2011).....	25
Tabela 2: Valores possíveis de kc (SHIGLEY, 1984)	26
Tabela 3: Diâmetros e áreas de roscas unificadas UNC e UNF (BUDYNAS, 2011).....	49
Tabela 4: Valores dos fatores de carga para eixos e árvores (ALBUQUERQUE, 1980)	61

SUMÁRIO

INTRODUÇÃO	9
1 METODOLOGIA.....	10
2 TENSÃO EM UM PONTO	12
2.1 ESTADO PLANO DE TENSÕES.....	13
2.2 TENSÕES PRINCIPAIS	14
2.3 CÓDIGO.....	15
3 TENSÕES COMBINADAS	17
3.1 CARGA AXIAL	17
3.2 TORÇÃO PURA	18
3.3 MOMENTO FLETOR.....	19
3.4 COMBINAÇÃO DE TENSÕES	20
3.5 TENSÕES PRINCIPAIS E TENSÃO DE CISALHAMENTO MÁXIMA NA COMBINAÇÃO DE TENSÕES.....	20
3.6 CÓDIGO.....	20
4. FADIGA	23
4.1 LIMITE DE RESISTÊNCIA À FADIGA	23
4.2 LIMITE DE RESISTÊNCIA À FADIGA DE ELEMENTOS DE MÁQUINAS	24
4.2.1 ACABAMENTO SUPERFICIAL (k_a)	25
4.2.2 TAMANHO DA PEÇA (k_b)	25
4.2.3 CONFIABILIDADE (k_c).....	25
4.2.4 TEMPERATURA (k_d).....	26
4.2.5 TIPO DE CARGA (k_e)	26
4.3 CÓDIGO.....	27
5. DIMENSIONAMENTO E ANÁLISE DE ELEMENTOS DE MÁQUINAS SUJEITO À FADIGA	29
5.1 TENSÕES FLUTUANTES	29
5.2 FATOR DE CONCENTRAÇÃO DE TENSÃO	30
5.3 RESISTÊNCIA À FADIGA SOB TENSÕES VARIÁVEIS	31
5.3.1 CRITÉRIO DE SORDERBERG	31
5.3.2 CRITÉRIO MODIFICADO DE GOODMAN	32
5.3.3 CRITÉRIO DE GERBER.....	33
5.3.4 CRITÉRIO ASME.....	33
5.3.5 CRITÉRIO DO ESCOAMENTO AMPLIADO	34
5.4 CRITÉRIO ESCOAMENTO DE von MISES	35
5.5 CÓDIGO.....	35
6. PARAFUSOS	44
6.1 DIÂMETRO, DIÂMETRO EFETIVO E ÁREA DE RESISTÊNCIA À TRAÇÃO.....	44
6.2 PRÉ-CARREGAMENTO.....	45

6.3 CÓDIGO.....	47
6.4 CARREGAMENTO EXCÊNTRICO	49
6.5 CÓDIGO.....	51
7. EIXOS E ÁRVORES	59
7.1 CÓDIGO.....	62
8. CHAVETAS E ESTRIAS	65
8.1 CÓDIGO.....	66
9. PROJETO E LUBRIFICAÇÃO DE MANCAIS DE DESLIZAMENTO	68
9.1 MASSA ESPECÍFICA E VISCOSIDADE DO LUBRIFICANTE	68
9.2 COEFICIENTE DE ATRITO	69
9.3 QUANTIDADE DE CALOR GERADA POR ATRITO	70
9.4 QUANTIDADE DE CALOR DISSIPADA.....	70
9.5 TEMPERATURA DE OPERAÇÃO DO ÓLEO LUBRIFICANTE	71
9.6 NÚMERO DE SOMMERFIELD	71
9.7 CÓDIGO.....	71
10. MANCAIS DE ROLAMENTO	73
10.1 VIDA E CONFIABILIDADE DE UM GRUPO DE ROLAMENTOS	73
10.2 CÓDIGO.....	74
10.3 SELEÇÃO DE UM ROLAMENTO.....	74
10.3.1 CARGA DINÂMICA EQUIVALENTE.....	74
10.3.2 CAPACIDADE DE CARGA DINÂMICA	75
10.3.3 CAPACIDADE DE CARGA ESTÁTICA REQUERIDA	75
10.4 CÓDIGO.....	75
CONCLUSÃO.....	77
BIBLIOGRAFIA	78

INTRODUÇÃO

A pandemia de COVID-19 trouxe desafios e mudanças significativas e sem precedentes em todos os âmbitos, incluindo o setor de educação superior. A Universidade do Estado do Rio de Janeiro (UERJ) iniciou o Período Acadêmico Emergencial em 14 de setembro de 2020, uma data histórica para a instituição que, pela primeira vez, atuou completamente de forma remota.

Até então, no curso de Engenharia Mecânica da UERJ, todas as disciplinas eram ministradas presencialmente e, para se adequar ao novo cenário, foi preciso alterar a metodologia de ensino para a nova realidade de isolamento social. Diversas ferramentas de ensino ganharam notoriedade para auxiliar alunos e professores, como plataformas de ambiente virtual e de videochamadas, por exemplo.

Nesse contexto, a disciplina Elementos de Máquinas I passou pelo mesmo desafio de se adequar ao ensino à distância (EAD). No curso presencial, um dos métodos utilizados para aperfeiçoar o aprendizado era uma apostila com conteúdo teórico-prático disponibilizada pelo professor, com diversos exercícios propostos. As dúvidas de resolução que surgiam eram mais facilmente sanadas em função da relação presencial entre aluno e professor, ou seja, a distância e outros fatores intrínsecos ao EAD, dificultaram esse processo. Notou-se, então, a necessidade de criar uma ferramenta para auxiliar os alunos na resolução dos exercícios, de modo a otimizar o aprendizado.

Assim, surge a idealização deste trabalho de conclusão de curso: automatizar os exercícios utilizados no processo de aprendizado da disciplina de Elementos de Máquinas I, de modo a estimular os alunos a desenvolverem estratégias para resolução a partir da análise crítica do resultado/resposta do exercício. Vale ressaltar que foram selecionados exercícios pertinentes de cada capítulo, visto que alguns deles são muito específicos ou de simples resolução, não sendo observado benefícios em sua automatização.

No decorrer deste trabalho serão mostrados os códigos utilizados, precedido de um resumo teórico, que permitiram a criação do dispositivo que se torna mais uma ferramenta disponível para o processo de aprendizagem, tanto no contexto remoto quanto no presencial.

1 METODOLOGIA

Para realizar a automatização dos exercícios foi necessária a escolha de uma linguagem de programação que atendesse todos os requisitos para realização dos processos. Neste caso, optou-se pelo uso do Python em função de alguns fatores, como sua alta aplicabilidade na atualidade, uma vez que há grande número de materiais documentados na internet, bem como alto número de colaboradores existentes. Levando isso em consideração, todo o código utilizado neste trabalho será compartilhado na plataforma GitHub pelo link https://github.com/Hugolm20/Projeto_graduacao_uerj_elemaq, uma comunidade usada pelos desenvolvedores de código para compartilhar seus algoritmos, tornando possível a discussão e o aprimoramento dos programas.

Além disso, essa linguagem possui grande número de bibliotecas, módulos e funções pré-programadas que ajudam a reduzir o tamanho do algoritmo, tornando-o conciso e facilmente interpretado para iniciantes e experts na área da programação. Uma das funcionalidades do Python é automatizar processos e foi elaborada para ser o mais próximo possível dos pseudocódigos, formas genéricas de escrever um algoritmo, que são mostrados logo quando se ensina lógica de programação.

Durante todo o trabalho, o leitor encontrará algumas imagens das linhas de códigos dos programas que representam como a linguagem de programação foi aplicada para encontrar as respostas dos exercícios da apostila.

No início de cada programa há o comando "import" para importar as bibliotecas. Estas são conjuntos de códigos criados previamente para auxiliar no processo de programação. Foram importadas a Math para realizar algumas operações matemáticas, Numpy para os cálculos com matrizes e a Decimal a fim de aplicar a precisão dupla.

Foi utilizado o uso de condicionais para tomadas de decisão, representadas na Figura 1, ferramenta essencial para que, durante a execução do código, uma ação fosse realizada de acordo com o dado de entrada.

```
if condição 1:  
    execute uma ação caso a condição se cumpra  
else:  
    caso contrário, execute outra ação
```

Figura 1: Estrutura condicional

Em outros pontos, foi necessário usar a estrutura de repetição com “while”, representada na Figura 2, que determina a execução do código repetidas vezes enquanto uma condição não é cumprida.

```
i = 0  
a = []  
  
while condição 1:  
    execute uma ação  
    i = i + 1
```

Figura 2: Exemplo de laço de repetição

Pode-se entender a relação entre esses comandos a partir da análise da Figura 2, em que o “*i*” representa uma variável, um valor que fica armazenado para ser utilizado futuramente, que, neste caso, funciona como um contador que começa em zero. A letra “*a*” também é uma variável, porém ela é uma lista que está vazia. Elas foram usadas, pois em alguns momentos ela representa uma matriz 3x3 e em outros um vetor com 2 ou 3 dimensões. Logo depois, é inserido o “while” com a condição e posteriormente as linhas de código são inseridas para executar uma série de ações. No final do código, sempre se adiciona “+1” ao contador para que o código continue rodando até que a condição estabelecida seja cumprida, ou seja, na primeira vez que o código é rodado, “*i*” é igual a zero, como no fim do código é adicionado um ao zero, então “*i*” passa a ser igual a 1. Assim, na segunda vez que o código for executado, $i = 1$ e no final mais um será adicionado e assim $i = 2$ e assim, sucessivamente, até que uma condição estabelecida seja cumprida.

Além dessas estruturas, na parte inicial do código, foram inseridos os dados de entrada onde é possível adicionar valores que ficarão armazenados nas variáveis para que depois sejam utilizados nos cálculos ao longo do código. Já na parte final, é possível ver os dados de saída que, em geral, apresentam os resultados dos exercícios da apostila.

Com essa visão geral do uso da linguagem Python no processo de automatização é possível seguir a leitura deste trabalho com o conhecimento necessário para compreender as linhas de códigos apresentadas. Para testá-las, é necessário o uso de um editor de código como Google Colaboratory, Jupyter, Visual Studio Code, etc.

2 TENSÃO EM UM PONTO

Segundo (SHIGLEY, 1984), fazer uma relação das tensões internas produzidas pelas cargas externas com a resistência de um elemento, é um trabalho difícil em projeto, já que normalmente disponibiliza-se apenas o limite de escoamento e os componentes de tensão. Ao longo deste trabalho será encontrado um desses problemas em que é necessário relacionar uma resistência com um elemento e encontrar o fator de segurança.

Neste capítulo, é apresentado o passo inicial da resolução deste problema dos exercícios que visam calcular os novos componentes de tensão quando o sistema sofre uma rotação em um determinado sentido, cálculo das tensões principais, planos de tensão e planos principais.

Os cálculos, cujo objetivo é calcular os componentes de tensão após a rotação, foram retirados do (SHIGLEY, 1984) e referenciados nas equações a seguir. A Eq. (2.1) calcula as tensões normais e a (2.2) a tensão cisalhante.

$$\sigma'_{x,y} = \frac{\sigma_x + \sigma_y}{2} \pm \frac{\sigma_x - \sigma_y}{2} \cos 2\phi \pm \tau_{xy} \sin 2\phi \quad (2.1)$$

$$\tau'_{xy} = - \frac{\sigma_x - \sigma_y}{2} \sin 2\phi \pm \tau_{xy} \cos 2\phi \quad (2.2)$$

Para entender a construção dessas equações, analisamos a Figura 3, que representa um paralelepípedo elementar com três tensões normais σ_x , σ_y , σ_z e seis tensões de cisalhamento representadas por: τ_{xy} , τ_{yx} , τ_{yz} , τ_{zy} , τ_{zx} e τ_{xz} , todas positivas. O primeiro índice ao lado da tensão de cisalhamento indica a coordenada normal à face do paralelepípedo e o segundo índice representa o eixo paralelo à tensão representada.

Este paralelepípedo está em equilíbrio estático, ou seja, todos seus momentos são iguais a zero, portanto, podemos admitir que: $\tau_{xy} = \tau_{yx}$, $\tau_{yz} = \tau_{zy}$ e $\tau_{zx} = \tau_{xz}$.

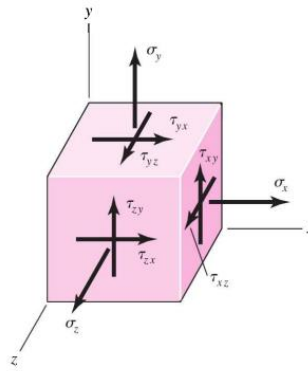


Figura 3: Paralelepípedo elementar (BUDYNAS, 2011)

São consideradas grandezas positivas as tensões normais que estão para a direita no eixo x (para fora do paralelepípedo elementar) e as tensões cisalhantes, que estão para cima na direção do eixo y . As tensões cisalhantes das facetas negativas também são consideradas grandezas positivas apesar de estarem atuando no sentido oposto.

2.1 ESTADO PLANO DE TENSÕES

Segundo (SHIGLEY, 1984) o estado plano de tensões é o mais usual e, nele, as tensões normais são tratadas como positivas ou negativas, enquanto as tensões cisalhantes são classificadas como sentido horário ou anti-horário.

O estado plano de tensões ocorre quando uma das superfícies tem tensões iguais a zero, ou seja, ela está sujeita a um estado de tensões plano ou biaxial, como representado na Figura 4 adaptada de (BUDYNAS, 2011). Supondo que a face com a seta apontando na direção z da Figura 3 esteja descarregada, temos que $\sigma_z = \tau_{xz} = \tau_{yz} = 0$. Dessa forma sobram apenas as tensões σ_y, σ_x e τ_{xy} .

Então o paralelepípedo é decomposto em um retângulo.

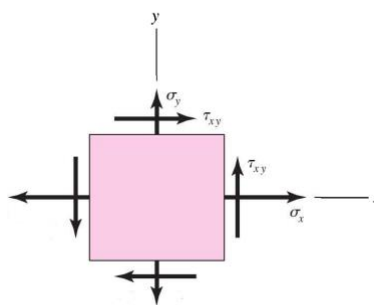


Figura 4: Estado plano de tensão (BUDYNAS, 2011)

Esses componentes de tensões restantes se alteram quando o sistema de referência é girado de um ângulo ϕ em torno do eixo x como representado na Figura 5.

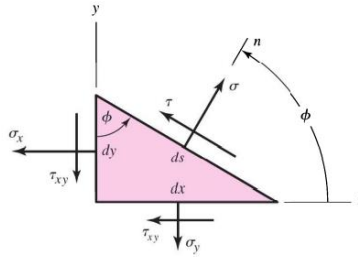


Figura 5: Novos componentes de tensão quando o sistema de referência é girado de um ângulo ϕ (BUDYNAS, 2011)

Quando se iguala a zero, em função do estado plano de tensões, a soma das forças causadas por todas as componentes das tensões, chega-se às Eqs. (2.1) e (2.2) utilizadas no exercício e demonstrada anteriormente no início do capítulo.

O código feito em Python utilizado nessa questão está representado a seguir:

```
import math
```

```
seno = math.sin(math.radians(2*angulo))
cosseno = math.cos(math.radians(2*angulo))

tensao_normal_linha_1 = ((tensao_normal_1 + tensao_normal_2)/2) +
(((tensao_normal_1-tensao_normal_2)/2) * (cosseno)) + (tensao_cisalhante_12 *
(seno))
tensao_normal_linha_2 = ((tensao_normal_1 + tensao_normal_2)/2) -
(((tensao_normal_1-tensao_normal_2)/2) * (cosseno)) - (tensao_cisalhante_12 *
(seno))
tensao_cisalhante_linha_12 = -(((tensao_normal_1 - tensao_normal_2)/2) * seno) +
(tensao_cisalhante_12 * cosseno)
```

Os dados de entrada são “*tensão_normal_1*” (σ_x), “*tensão_normal_2*” (σ_y), “*tensão_cisalhante_12*” (τ_{xy}) e “*ângulo*” (θ).

2.2 TENSÕES PRINCIPAIS

A partir das equações usadas no exercício da seção anterior, chega-se à equação da tangente de 2ϕ que retorna dois valores: o primeiro, ϕ_n , representa o ângulo de rotação do sistema de referência cuja tensão σ será máxima. O segundo, ϕ_t , o ângulo de rotação para o qual a tensão tangencial τ é máxima. Sendo o primeiro representado pela Eq. (2.3) a seguir:

$$\operatorname{tg} 2\phi_n = \frac{2\tau_{xy}}{\sigma_x - \sigma_y} \quad (2.3)$$

De forma análoga, a Eq. (2.4) retorna o outro valor:

$$\operatorname{tg} 2\phi_t = \frac{\sigma_x - \sigma_y}{2\tau_{xy}} \quad (2.4)$$

Substituindo a primeira Eq. (2.3) na (2.1), encontra-se:

$$\sigma_{máx,min} = \frac{\sigma_x + \sigma_y}{2} \pm \sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \tau_{xy}^2} \quad (2.5)$$

De forma semelhante, substituindo a Eq. (2.4) na Eq. (2.2), temos:

$$\tau_{máx} = \sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \tau_{xy}^2} \quad (2.6)$$

Por fim, para calcular a tensão média, usa-se:

$$\sigma_m = \frac{\sigma_x + \sigma_y}{2} \quad (2.7)$$

As equações acima foram retiradas do (SHIGLEY, 1984).

2.3 CÓDIGO

Ao longo deste capítulo, pode-se compreender como as Eqs. (2.3), (2.4), (2.5), (2.6) e (2.7) foram estabelecidas teoricamente. A partir disso, mostra-se o código desenvolvido para a automatização dos exercícios cujo assunto é o cálculo das tensões principais e tensão de cisalhamento máxima da apostila.

```

import math

# letra a
tan_2_teta_n = (2 * tensao_cisalhante_12) / (tensao_normal_1 - tensao_normal_2)
dois_teta_n = math.degrees(math.atan(tan_2_teta_n))
 $\theta_{n1}$  = dois_teta_n / 2
 $\theta_{n2}$  =  $\theta_{n1}$  + 90

# letra b
tensao_normal_max = ((tensao_normal_1 + tensao_normal_2) / 2) +
math.sqrt(((tensao_normal_1 - tensao_normal_2) / 2)**2 + tensao_cisalhante_12**2)
tensao_normal_min = ((tensao_normal_1 + tensao_normal_2) / 2) -
math.sqrt(((tensao_normal_1 - tensao_normal_2) / 2)**2 + tensao_cisalhante_12**2)

# letra c
tan_2_teta_t = -((tensao_normal_1 - tensao_normal_2) / (2 * tensao_cisalhante_12))
dois_teta_t = math.degrees(math.atan(tan_2_teta_t))
 $\theta_{t1}$  = dois_teta_t / 2
 $\theta_{t2}$  =  $\theta_{t1}$  + 90

# letra d
tensao_cisalhante_max = math.sqrt(((tensao_normal_1 - tensao_normal_2) / 2)**2 +
tensao_cisalhante_12**2)

# letra e
tensao_media = ((tensao_normal_1 + tensao_normal_2) / 2)

```

Os dados de entrada são:

- $tensao_normal_1$ (σ_x)
- $tensao_normal_2$ (σ_y)
- $tensao_cisalhante_12$ (τ_{xy})

Os dados de saída são:

- ϕ_{n1} e ϕ_{n2} , que representam os planos principais
- $tensao_normal_max$ ($\sigma_{máx}$)
- $tensao_normal_min$ ($\sigma_{mín}$)
- ϕ_t , em que a tensão tangencial é máxima
- $tensao_cisalhante_max$ ($\tau_{máx}$)
- $tensão_media$ (σ_m)

3 TENSÕES COMBINADAS

Durante a análise de uma estrutura ou um elemento de uma máquina, percebe-se que há uma ou várias combinações de tensões e cargas atuantes, como pode ser visto na Figura 6.

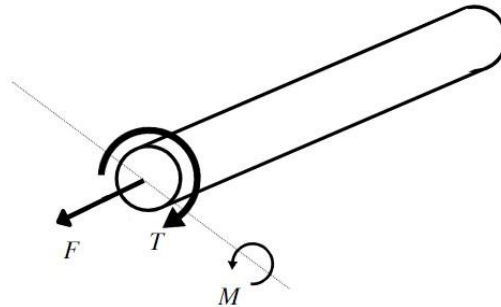


Figura 6: Corpo cilíndrico sujeito a esforços combinados.

Nesta etapa do trabalho, os principais exercícios automatizados da apostila fazem parte da seção cujo assunto é: Tensões principais e tensão de cisalhamento máxima na combinação de tensões. Todos esses exercícios selecionados têm em comum o objetivo de encontrar valores das tensões a partir de um elemento que está sujeito a cargas e momentos de flexão e torção. Vale ressaltar que as equações desta seção podem ser encontradas em diversos livros de Resistência de Materiais, mas, para este trabalho, foram retiradas do (LA PLATA, 2022). Além disso, neste projeto, serão analisados apenas cargas e tensões aplicadas em eixos cilíndricos com diâmetro interno e externo.

3.1 CARGA AXIAL

A primeira equação que é necessário ter conhecimento é a da tensão normal gerada quando o elemento está sujeito a uma carga axial:

$$\sigma = \frac{F}{S} \quad (3.1)$$

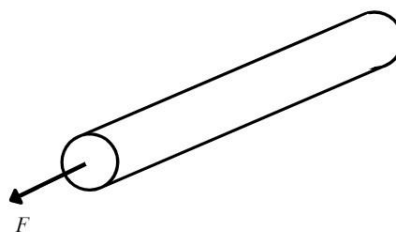


Figura 7: Elemento submetido a uma carga axial F

Sendo F a carga axial e S é a área da seção transversal do elemento, que no caso do cilindro, é dada por:

$$S = \frac{\pi}{4}(d_0^2 - d_i^2) \quad (3.2)$$

Onde, d_0 é o diâmetro externo e d_i o diâmetro interno. Substituindo a Eq. (3.2) na Eq. (3.1), tem-se:

$$\sigma = \frac{4F}{\pi \cdot d_0^2 (1 - k^2)} \quad (3.3)$$

Sabe-se que k , a relação entre os diâmetros, pode ser calculado a partir da Eq. (3.4), demonstrada a seguir:

$$k = \frac{d_i}{d_0} \quad (3.4)$$

3.2 TORÇÃO PURA

Quando o eixo está sujeito a um momento de torção, é gerada uma tensão de cisalhamento (τ) definida por:

$$\tau = \frac{T \cdot c}{J} \quad (3.5)$$

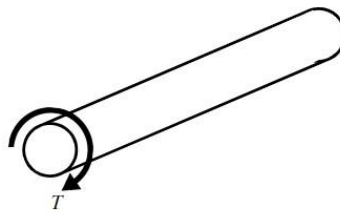


Figura 8: Elemento submetido a um momento de torção T puro

Sendo J o momento polar de inércia da área da seção transversal definido por:

$$J = \frac{\pi}{32}(d_0^4 - d_i^4) \quad (3.6)$$

Sendo c , raio externo do eixo, calculado a partir da Eq. a seguir:

$$c = \frac{d_0}{2} \quad (3.7)$$

Usando a relação entre os diâmetros k e substituindo as Equações (3.6) e (3.7) na Eq. (3.5), tem-se a Eq. (3.8):

$$\tau = \frac{16 T}{\pi \cdot d_0^3 (1-k^4)} \quad (3.8)$$

3.3 MOMENTO FLETOR

Quando o eixo está submetido a um momento fletor, a tensão normal (σ) relacionada ao momento fletor é calculada a partir da Eq. (3.11), que correlaciona as Eqs. (3.9) e (3.10), definidas a seguir.

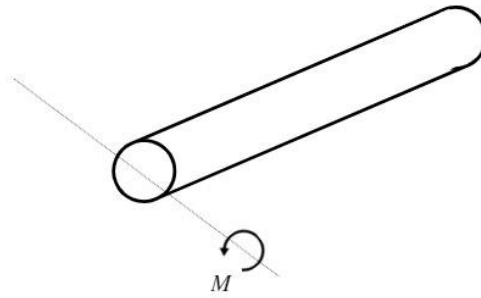


Figura 9: Elemento submetido a um momento de flexão M puro

$$\sigma = \frac{M \cdot c}{I} \quad (3.9)$$

$$I = \frac{\pi}{64} (d_0^4 - d_i^4) \quad (3.10)$$

$$\sigma = \frac{32 M}{\pi \cdot d_0^3 (1-k^4)} \quad (3.11)$$

3.4 COMBINAÇÃO DE TENSÕES

A Figura 6 mostrada no início desse capítulo representa um cenário em que todas as tensões definidas ao longo do capítulo estão atuando, considerando que o material permanece em regime elástico, permitindo o uso do princípio da superposição para encontrar as resultantes.

Nesse caso, a tensão normal é calculada pelo somatório dela sob efeito da carga axial com a do momento de flexão, representada pela Eq. (3.12), enquanto a tensão de cisalhamento permanece sendo calculada pela Eq. (3.8).

$$\sigma = \frac{4F}{\pi \cdot d_0^2 (1-k^2)} + \frac{32M}{\pi \cdot d_0^2 (1-k^4)} \quad (3.12)$$

3.5 TENSÕES PRINCIPAIS E TENSÃO DE CISALHAMENTO MÁXIMA NA COMBINAÇÃO DE TENSÕES

No caso da combinação de tensões, existe apenas um componente de tensão normal e apenas um componente para tensão cisalhante, como mostra a Figura 10. Desta maneira, as tensões principais e de cisalhamento serão dadas por equações parecidas com as Eqs. (2.5) e (2.6), entretanto haverá apenas um elemento de cada.

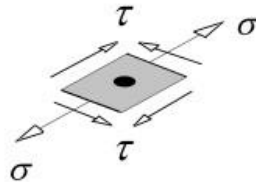


Figura 10: Tensão plana de um eixo sujeito a esforços combinados (LA PLATA, 2022)

3.6 CÓDIGO

A partir da construção e correlação das equações anteriores, tem-se a base teórica para compreender o código usado para solucionar os problemas de L invertidos, como ilustrado na Figura 11, que tratam do assunto de tensões principais e tensão de cisalhamento máxima na combinação de tensões. A seguir, é exibido como exemplo, a automatização do exercício cujos dados de entrada são hipotéticos.

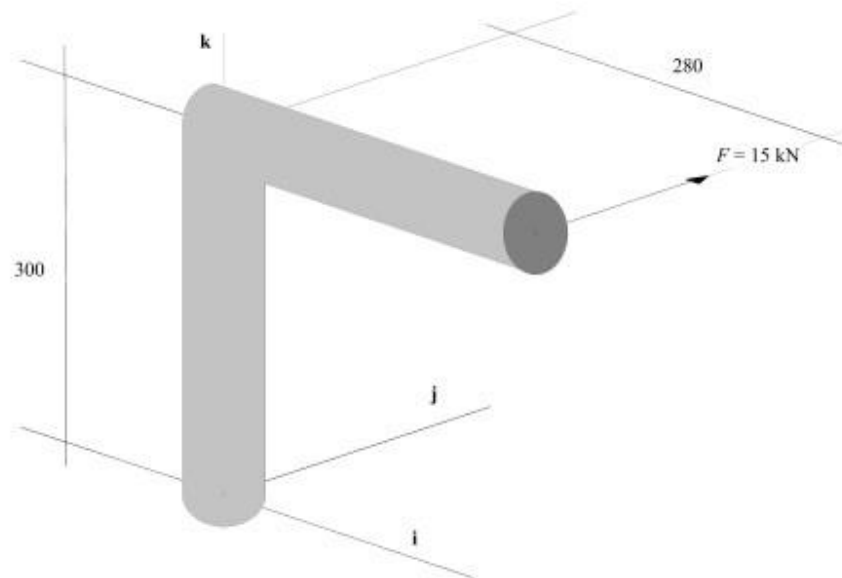


Figura 11: Representação da estrutura de L invertido com suas dimensões e uma carga axial F aplicada (LA PLATA, 2022)

```
import numpy as np
import math

d = 0.08
matriz_posicao = [[0.280, 0, 0.300]]
matriz_forca = [[0, 15, 0]]
matriz_momento = ''

i = 0
a = []

while i < len(matriz_posicao):
    prod = np.cross(matriz_posicao[i], matriz_forca[i])
    prod = a.append(prod)
    prod = sum(a)
    i = i + 1
if matriz_momento == '':
    prod = list(prod)
else:
    matriz_momento = np.sum(matriz_momento, axis=0)
    prod = prod + matriz_momento
    prod = list(prod)

matriz_forca = np.sum(matriz_forca, axis=0)

t = prod[2]
m = math.sqrt(((prod[0])**2) + ((prod[1])**2))

tensao_cisalhante = ((16 * t) / (3.14 * (d**3)))
tensao_normal = ((32 * m) / (3.14 * (d**3)))
tensao_cisalhante_max = (math.sqrt(((tensao_normal / 2)**2) +
(tensao_cisalhante**2))) * 10**-3
```

No início do código foram importadas as bibliotecas “math” para realização de algumas operações matemáticas e a “numpy”, utilizada para trabalhar com listas. Os dados de entrada são:

- diâmetro (d)
- *matriz_posicao*, que representa a posição x, y e z da força aplicada.
- *matriz_forca*, a força aplicada, onde cada índice da lista representa a dimensão

Nesse exercício, foi aplicada uma força de 15 kN na direção y . Ambas matrizes deverão ser multiplicadas uma pelas outra através do produto vetorial para achar o vetor momento, tendo como resultado $-4.5 \mathbf{i} + 0 \mathbf{j} + 4.20 \mathbf{k}$. Caso haja um momento aplicado na estrutura, ele é somado ao cálculo anterior automaticamente.

Por fim, os dados de saída são:

- torção(t), o terceiro elemento do vetor
- momento de flexão (m). a raiz quadrada da soma do primeiro e segundo elemento
- tensão cisalhante (*tensao_cisalhante*)
- tensão normal (*tensao_normal*)
- tensão cisalhante máxima (*tensao_cisalhante_max*).

4. FADIGA

A falha por fadiga de um elemento estrutural ou de máquinas ocorre sob a ação de tensões que se repetem ou flutuam por um número muito grande de vezes. Segundo (BUDYNAS, 2011), uma análise mais cuidadosa revela que as tensões máximas verdadeiras estavam bem abaixo da resistência última do material (limite de resistência) e muito frequentemente até abaixo da resistência ao escoamento (limite de escoamento).

A fratura por esse tipo de falha se inicia com microtrincas que se propagam e aumentam de tamanho devido as tensões aplicadas repetidamente causando deformação plástica até o rompimento do elemento. Ocorrem em regiões de descontinuidade do material, também chamadas de concentradores de tensão. Podem ser em um rasgo de chaveta, irregularidades em decorrência da usinagem etc.

4.1 LIMITE DE RESISTÊNCIA À FADIGA

O objetivo nesta seção é calcular o limite de resistência à fadiga (σ_n) que é determinado a partir de ensaios que submetem corpos de prova a cargas de flexão alternada ilustrado na Figura 12. Então os dados são computados e colocados em um gráfico como o da Figura 13.

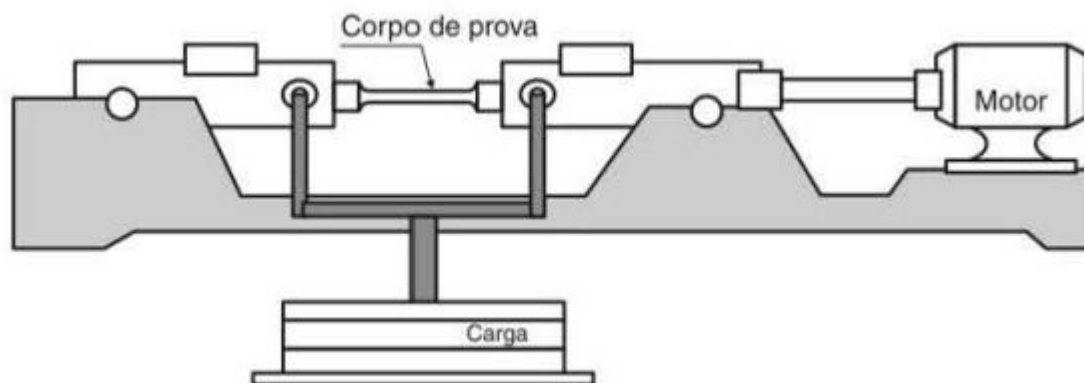


Figura 12: Máquina de flexões alternadas utilizadas no ensaio de fadiga (GARCIA, 2012)

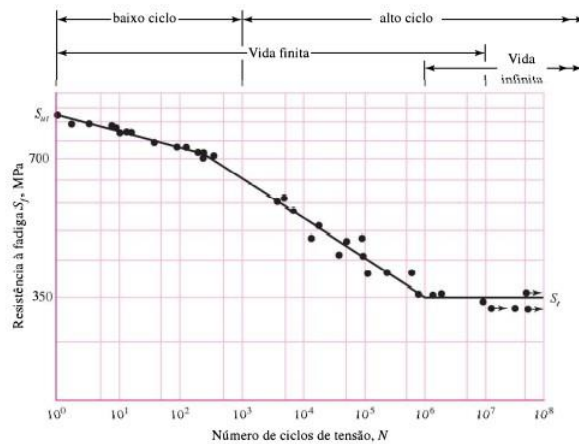


Figura 13: Dados do ensaio computados em um gráfico (BUDYNAS, 2011)

É importante perceber no gráfico que a resistência à fadiga está representada por S_f , porém, neste trabalho, a variável é representada por σ_n . Também há um ponto onde os números de ciclos se estendem e passam a ter valores bem próximo um dos outros sendo possível traçar uma linha horizontal, ocorrendo um “joelho”. A partir daí pode-se tirar o valor do Limite de resistência à fadiga do material.

No caso deste trabalho, os exercícios automatizados utilizam apenas materiais ferrosos, portanto, de acordo com (BUDYNAS, 2011), a equação condicional a seguir (4.1) classifica o limite de resistência à fadiga a partir do valor da tensão e o limite de ruptura do material (σ_r).

$$\sigma'_n = \begin{cases} 0,50\sigma_r & \text{se } \sigma_r \leq 1400 \text{ MPa} \\ 700 \text{ MPa} & \text{se } \sigma_r > 1400 \text{ MPa} \end{cases} \quad (4.1)$$

4.2 LIMITE DE RESISTÊNCIA À FADIGA DE ELEMENTOS DE MÁQUINAS

O σ'_n se refere ao valor calculado para corpos de prova padronizados. Quando levamos o caso para peças reais, ele deve ser corrigido pelo seguinte cálculo:

$$\sigma_n = k_a \cdot k_b \cdot k_c \cdot k_d \cdot k_e \cdot k_f \cdot \sigma'_n \quad (4.2)$$

Onde σ_n é o Limite de resistência à fadiga da peça e σ'_n o Limite de resistência à fadiga do corpo de prova padrão sujeito à flexão alternada. Já os k 's são fatores que levam em consideração alguns fatores, onde k_a leva em consideração o acabamento superficial da peça,

k_b considera o tamanho da peça, k_c , a confiabilidade, k_d a temperatura, k_e o tipo de carga e k_f fator de efeitos diversos. Para o cisalhamento a equação é a mesma, porém a variável é τ_n .

4.2.1 ACABAMENTO SUPERFICIAL (k_a)

Para o cálculo do k_a , segundo (BUDYNAS, 2011), temos a seguinte equação:

$$k_a = a \cdot \sigma_r^b \quad (4.3)$$

Onde σ_r é o limite de ruptura em MPa, a e b são constantes obtidas a partir da Tabela 1, a seguir.

Tabela 1: Valores de a e b para determinação do k_a (BUDYNAS, 2011)

Acabamento Superficial	a	b
Retificado	1,58	-0,085
Usinado ou laminado a frio	4,51	-0,265
Laminado a quente	57,7	-0,718
Forjado	272	-0,995

4.2.2 TAMANHO DA PEÇA (k_b)

O limite de resistência à fadiga é inversamente proporcional ao tamanho da peça, pois quanto maior a peça, maior a quantidade de defeitos superficiais que podem ser concentradores de tensão.

Para o cálculo de k_b , segundo (BUDYNAS, 2011), considera-se que:

$$k_b = \begin{cases} \left(\frac{d}{7,62}\right)^{-0,107} & \text{se } 2,79 \text{ mm} \leq d \leq 51,00 \text{ mm} \\ 1,51 \cdot d^{-0,157} & \text{se } 51,00 < d \leq 254,00 \text{ mm} \end{cases} \quad (4.4)$$

Para cargas axiais, temos que $k_b = 1$. Nos casos em que não é dado o valor do diâmetro, recomenda-se usar $k_b = 0,85$.

4.2.3 CONFIABILIDADE (k_c)

O valor da confiabilidade é obtido por meio da análise da Tabela 2, adaptada de (SHIGLEY, 1984), mostrada a seguir.

Tabela 2: Valores possíveis de k_c (SHIGLEY, 1984)

Confiabilidade (R)	Fator de Confiabilidade (k_c)
0,50	1,000
0,90	0,897
0,95	0,868
0,99	0,814
0,999	0,753
0,9999	0,702
0,99999	0,659
0,999999	0,620
0,9999999	0,584
0,99999999	0,551
0,999999999	0,520

4.2.4 TEMPERATURA (k_d)

O fator de correção k_d é dado pela Eq. 4.5, segundo (BUDYNAS, 2011), em que $37 \leq T_c \leq 540^\circ\text{C}$.

$$k_d = 0,9877 + 0,6507(10^{-3}) T_C - 0,3414(10^{-5}) T_C^2 + 0,5621(10^{-8}) T_C^3 - 6,246(10^{-12}) T_C^4 \quad (4.5)$$

4.2.5 TIPO DE CARGA (k_e)

Segundo (SHIGLEY, 1984) é recomendado usar:

$$k_e = \begin{cases} 1,00 & \text{para flexão} \\ 0,85 & \text{para carga axial} \\ 0,577 & \text{para torção} \end{cases} \quad (4.6)$$

4.2.6 FATORES DIVERSOS (k_f)

Segundo (SHIGLEY, 1984), quando nada for estabelecido, é recomendado tomar o valor de k_f igual a 1, pois esses outros fatores são variantes e difíceis de calcular. Eles dependem das tensões residuais que podem tanto melhorar quanto afetar o limite de resistência

4.3 CÓDIGO

A seguir, mostra-se o código utilizado para automatizar o exercício.

```
#ka
if aco == 'forjado':
    a = 272
    b = -0.995
    ka = a * (tensao_ruptura ** b)
elif aco == 'laminado_quente':
    a = 57.7
    b = -0.718
    ka = a * (tensao_ruptura ** b)
elif aco == 'usinado_ou_laminado_frio':
    a = 4.51
    b = -0.265
    ka = a * (tensao_ruptura ** b)
elif aco == 'retificado':
    a = 1.58
    b = -0.085
    ka = a * (tensao_ruptura ** b)
elif aco == 'espelhado':
    a = 1
    b = 0
    ka = a * (tensao_ruptura ** b)

#kb
if d == '':
    kb = 0.85
elif 2.79 <= d <= 51:
    kb = (d / 7.62)**-0.107
elif 51 < d <= 254:
    kb = 1.51 * (d**-0.157)

#kc
if conf == 50:
    kc = 1
elif conf == 90:
    kc = 0.897
elif conf == 95:
    kc = 0.868
elif conf == 98:
    kc = 0.836
elif conf == 99:
    kc = 0,814
elif conf == 99.99:
    kc = 0.702
elif conf == 99.9999:
    kc = 0.620
elif conf == 99.99999:
    kc = 0.584

#kd
```

```

if 37 <= t <= 540:
    kd = 0.9877 + (0.6507*(10**-3)*t) - (0.3414*(10**-5)*(t**2)) + (0.5621*(10**-
8)*(t**3)) - (6.246*(10**-12)*(t**4))
else:
    kd = 1
#ke
if carga == 'flexao':
    ke = 1
elif carga == 'axial':
    ke = 0.85
elif carga == 'torcao':
    ke = 0.577
#kf
kf = 1
lim_res = ka * kb * kc * kd * ke * kf * (0.5 * tensao_ruptura)

```


5. DIMENSIONAMENTO E ANÁLISE DE ELEMENTOS DE MÁQUINAS SUJEITO À FADIGA

Alguns critérios de resistência à fadiga serão expostos neste capítulo visando relacionar cargas aplicadas com os fatores de segurança. Desta forma, junto com o limite de resistência à fadiga, se terá conceitos necessários para resolver o exercício de projeto, que visa projetar ou analisar elementos estruturais ou de máquinas submetidos a cargas variantes com o tempo, deste capítulo e os outros exercícios que o compõe.

5.1 TENSÕES FLUTUANTES

Segundo (SHIGLEY, 1984), em um projeto é necessário calcular resistência de elementos de máquinas ou estruturas com cargas que possuem tensões variáveis, mas não completamente alternadas, podendo variar e não cruzar o valor zero. Na Figura 14 há diversos casos possíveis de relação tensão-tempo.

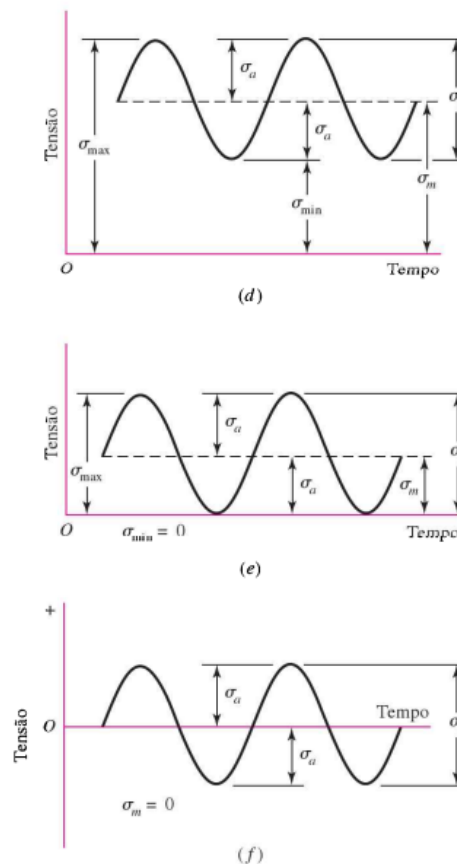


Figura 14: Relações de tempo-tensão. (d) Tensão variada (e) Tensão repetida (f) Tensão alternada (BUDYNAS, 2011)

σ_{min} = tensão mínima

$\sigma_{m\acute{a}x}$ = tensão máxima

σ_v = componente de amplitude

σ_m = tensão média

σ_r = variação de tensão

De modo que, pode-se calcular σ_m e σ_v a partir das equações a seguir.

$$\sigma_m = \frac{\sigma_{m\acute{a}x} + \sigma_{min}}{2} \quad (5.1)$$

$$\sigma_v = \frac{\sigma_{m\acute{a}x} - \sigma_{min}}{2} \quad (5.2)$$

5.2 FATOR DE CONCENTRAÇÃO DE TENSÃO

Concentradores de tensão são descontinuidades, como defeitos na soldagem, variações de dimensão e cantos vivos, que promovem aumento da tensão no local.

Quando calculamos as tensões através de suas fórmulas com momento flexão e torção, consegue-se ter a estimativa da tensão nominal (σ_0), porém, essas grandezas não informam a magnitude da tensão real (σ_{real}) em todos os pontos do elemento. Também é possível demonstrar que a tensão real próxima a uma descontinuidade é maior do que a tensão nominal ($\sigma_{nominal}$), que pode ser obtida de forma numérica ou analítica.

A relação entre essas duas tensões é dada por:

$$\sigma_{real} = k \cdot \sigma_0 \quad (5.3)$$

Onde, de acordo com a apostila (LA PLATA, 2022), k é um número real denominado fator de concentração. Quando ele é obtido da mesma forma que a tensão real, ele recebe o nome de fator teórico de tensão, representado por k_t . Em materiais dúcteis, k_t torna maior a estimativa da tensão real e, por esse motivo, existe o fator prático de concentração de tensão k_p , que, segundo (BUDYNAS, 2011) pode ser calculado da seguinte forma:

$$k_p = 1 + q \cdot (k_t - 1) \quad (5.4)$$

Nessa equação, q é o índice de sensibilidade ao entalhe. Vale ressaltar que valores de q e k_t podem ser determinados experimentalmente, mas neste trabalho são apresentados como dados de entrada dos exercícios.

5.3 RESISTÊNCIA À FADIGA SOB TENSÕES VARIÁVEIS

Para esta seção, a apostila oferece diversos critérios de resistência à fadiga que, em resumo, representam relações que surgem de semelhanças de triângulos entre tensão média (σ_m), variável (σ_v), limite de escoamento (σ_e), limite de resistência à fadiga (σ_n) e fator de segurança (N). O gráfico do critério de Sorderberg representado na Figura 15 explica graficamente essas relações. Além disso, esses gráficos possuem uma região de segurança contra o colapso por fadiga que muda de acordo com o critério.

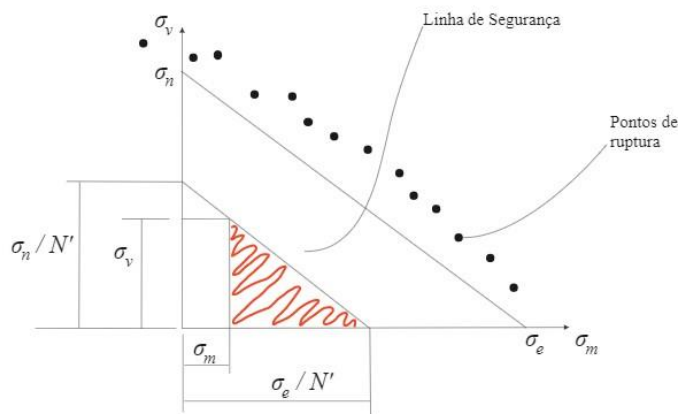


Figura 15: Diagrama de Sorderberg (LA PLATA, 2022)

Nas próximas subseções serão citados os critérios de resistência à fadiga que estão na apostila, (LA PLATA, 2022), e foram usados nos códigos dos exercícios deste capítulo.

5.3.1 CRITÉRIO DE SORDERBERG

Conforme ilustrado na Figura 15, o Critério de Sorderberg usa uma linha reta para definir uma região segura contra colapso por fadiga que vai de σ_n a σ_e . Suas equações são dadas a seguir:

$$\frac{\sigma_v}{\sigma_n} + \frac{\sigma_m}{\sigma_e} = \frac{1}{N'} \quad (5.5)$$

$$\sigma = \sigma_m + \sigma_v \frac{\sigma_e}{\sigma_n} \quad (5.6)$$

$$\tau = \tau_m + \tau_v \frac{\tau_e}{\tau_n} \quad (5.7)$$

$$N = \frac{\sigma_e}{(\sigma^2 + 3\tau^2)^{\frac{1}{2}}} \quad (5.8)$$

5.3.2 CRITÉRIO MODIFICADO DE GOODMAN

Já no Critério Modificado de Goodman, a linha de colapso liga σ_n a σ_r como mostra a Figura 16.

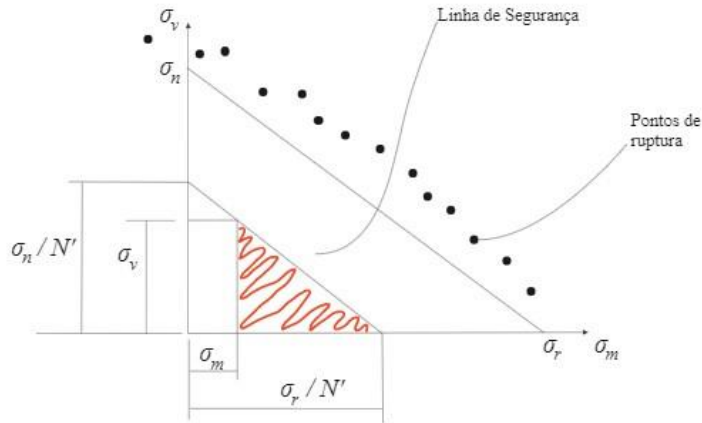


Figura 16: Critério Modificado de Goodman (LA PLATA, 2022)

$$\frac{\sigma_v}{\sigma_n} + \frac{\sigma_m}{\sigma_e} = \frac{1}{N'} \quad (5.9)$$

$$\sigma = \sigma_e \left(\frac{\sigma_v}{\sigma_n} + \frac{\sigma_m}{\sigma_e} \right) \quad (5.10)$$

$$\tau = \tau_e \left(\frac{\tau_v}{\tau_n} + \frac{\tau_m}{\tau_e} \right) \quad (5.11)$$

5.3.3 CRITÉRIO DE GERBER

De acordo com a Figura 17, a linha de colapso se forma com um arco de parábola que une os pontos σ_n e σ_r .

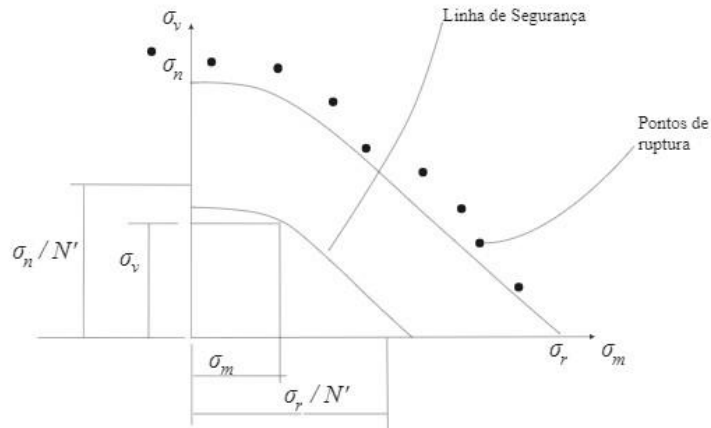


Figura 17: Critério de Gerber (arco de parábola) (LA PLATA, 2022)

$$\frac{N' \cdot \sigma_v}{\sigma_n} + \left(\frac{N' \cdot \sigma_m}{\sigma_r} \right)^2 = 1 \quad (5.12)$$

$$\sigma^2 - \frac{\sigma_e \cdot \sigma_v}{\sigma_n} \sigma - \left(\frac{\sigma_e \cdot \sigma_m}{\sigma_r} \right)^2 = 0 \quad (5.13)$$

$$\tau^2 - \frac{\tau_e \cdot \tau_v}{\tau_n} \tau - \left(\frac{\tau_e \cdot \tau_m}{\tau_r} \right)^2 = 0 \quad (5.14)$$

5.3.4 CRITÉRIO ASME

No Critério ASME a linha de colapso é determinada pelo arco de elipse entre σ_n e σ_e conforme ilustrado na Figura 18.

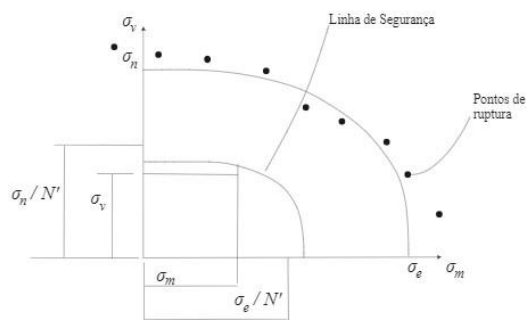


Figura 18: Critério ASME (LA PLATA, 2022)

$$\left(\frac{N' \cdot \sigma_v}{\sigma_n}\right)^2 + \left(\frac{N' \cdot \sigma_m}{\sigma_r}\right)^2 = 1 \quad (5.15)$$

$$\tau = \left[\left(\frac{\sigma_e \cdot \sigma_v}{\sigma_n} \right)^2 + \sigma_m^2 \right]^{\frac{1}{2}} \quad (5.16)$$

$$\tau = \left[\left(\frac{\tau_e \cdot \tau_v}{\tau_n} \right)^2 + \tau_m^2 \right]^{\frac{1}{2}} \quad (5.17)$$

5.3.5 CRITÉRIO DO ESCOAMENTO AMPLIADO

No Critério do escoamento Ampliado, como mostra a Figura 19, a linha que define a região segura é uma reta que passa pelos pontos $(\sigma_e, 0)$ e $(0, \sigma_e)$.

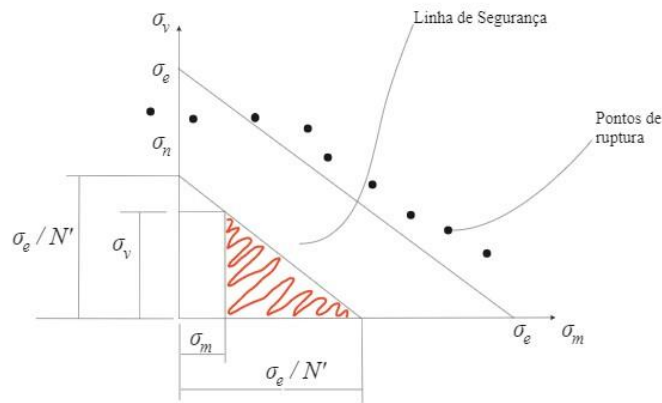


Figura 19: Critério do Escoamento Ampliado (LA PLATA, 2022)

$$\sigma_v + \sigma_m = \frac{\sigma_e}{N} \quad (5.18)$$

$$\sigma = \sigma_m + \sigma_v \quad (5.19)$$

$$\tau = \tau_m + \tau_v \quad (5.20)$$

5.4 CRITÉRIO ESCOAMENTO DE von MISES

Um ponto em comum em todos os critérios de fadiga apresentados acima é a Eq. (5.8), utilizada para encontrar o fator de segurança. É oriunda da energia de distorção máxima por unidade de volume (u_d) e das equações das tensões principais em um estado de tensão. Em resumo, u_d é a divisão do trabalho necessário para distorcer o elemento sobre o volume da peça.

Segundo (LA PLATA, 2022), por esse critério o u_d necessário para atingir o escoamento do material, é uma constante para esse material. Dessa forma, não importa se o corpo está sob efeito de uma carga ou cargas combinadas, permanece constante a energia de distorção por unidade de volume para atingir o escoamento ($u_d)_e$, que não será deformado enquanto $u_d < (u_d)_e$.

Assim, a partir dessas considerações, conclui-se que é possível chegar à Eq. (5.8) usando as equações de u_d , $(u_d)_e$ e tensões normais principais.

5.5 CÓDIGO

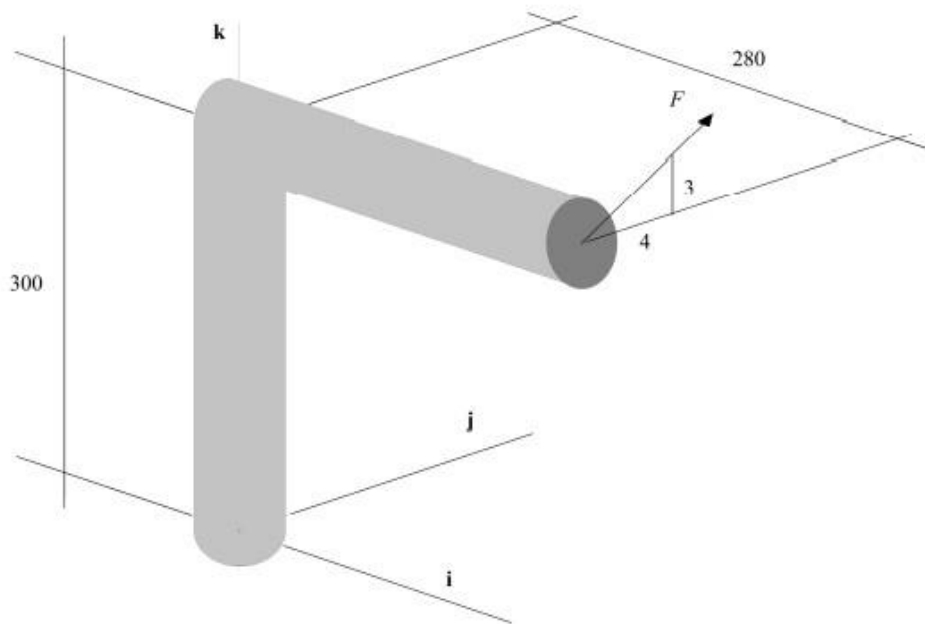


Figura 20: Ilustração da figura utilizada no exercício de projeto do capítulo 5. Projeto de uma peça sujeita a um esforço variável (LA PLATA, 2022)

No código deste exercício são utilizados os cálculos anteriormente automatizados neste trabalho, como o limite de resistência à fadiga, os valores de k , o fator de concentração de tensão e o produto vetorial para encontrar os momentos de flexão e torção. Por conta disso, nesta seção,

será oportunamente apresentado apenas parte do código envolvido, sendo possível acessar o repositório no GitHub para consultar o código na íntegra.

```
def tensao_normal_max(d):
    return (kp * ((Decimal(32) * m_max) / (Decimal(pi) * (d**Decimal(3))))) +
    (kp * ((Decimal(4) * f_max) / (Decimal(pi) * (d**Decimal(2)))))

def tensao_normal_min(d):
    return (kp * ((Decimal(32) * m_min) / (Decimal(pi) * (d**Decimal(3))))) +
    (kp * ((Decimal(4) * f_min) / (Decimal(pi) * (d**Decimal(2)))))

def tensao_cisalhante_max(d):
    return (kp * ((Decimal(16) * t_max) / (Decimal(pi) * (d**Decimal(3)))))

def tensao_cisalhante_min(d):
    return (kp * ((Decimal(16) * t_min) / (Decimal(pi) * (d**Decimal(3)))))

def tensao_normal_var(d):
    return (tensao_normal_max(d) - tensao_normal_min(d)) / Decimal(2)

def tensao_normal_media(d):
    return (tensao_normal_max(d) + tensao_normal_min(d)) / Decimal(2)

def tensao_cisalhante_media(d):
    return (tensao_cisalhante_max(d) + tensao_cisalhante_min(d)) / Decimal(2)

def tensao_cisalhante_var(d):
    return (tensao_cisalhante_max(d) - tensao_cisalhante_min(d)) / Decimal(2)
```

Nesta parte foram feitos os cálculos das tensões normais máximas, mínima, variável e média e o mesmo tipo de tensão para as cisalhantes. Nas fórmulas desses cálculos, a variável d foi colocada em evidência para que, no final, ela seja calculada. Porém, antes disso, é necessário utilizar algum método de fadiga para que seja possível encontrar a tensão normal e cisalhante. Para que a pessoa que for utilizar o código possa escolher o tipo que será utilizado, foi criada a variável critério que pode receber números de 1 a 5 sendo:

1. Critério de Sorderberg;
2. Critério modificado de Goodman;
3. Critério de Gerber;
4. Critério ASME;
5. Critério do Escoamento Ampliado.

```
#critério de sorderberg
if criterio == 1:
    def tensao_normal(d):
        return tensao_normal_media(d) + (tensao_normal_var(d) * (lim_esc/li
m_normal_res))
```



```

def tensao_cisalhante(d):
    return tensao_cisalhante_media(d) + (tensao_cisalhante_var(d) * (tensao_cisalhante_esc/lim_cisalhante_res))

#critério modificado de goodman
elif criterio == 2:
    def tensao_normal(d):
        return lim_esc * ((tensao_normal_var(d) / lim_normal_res) + (tensao_normal_media(d) / Decimal(tensao_ruptura)))

    def tensao_cisalhante(d):
        return tensao_cisalhante_esc * ((tensao_cisalhante_var(d) / lim_cisalhante_res) + (tensao_cisalhante_media(d) / (Decimal(tensao_ruptura)*Decimal(0.8))))

#critério gerber
elif criterio == 3:
    def tensao_normal(d):
        a1 = Decimal(1)
        b1 = (lim_esc * tensao_normal_var(d)) / lim_normal_res
        c1 = ((lim_esc * tensao_normal_media(d)) / Decimal(tensao_ruptura))
        delta1 = b1*b1 - (Decimal(4)*a1*(-c1))
        return (Decimal(-b1) - Decimal(math.sqrt(delta1)) ) / (Decimal(2)*a1)

    def tensao_cisalhante(d):
        a2 = Decimal(1)
        b2 = (tensao_cisalhante_esc * tensao_cisalhante_var(d)) / lim_cisalhante_res
        c2 = ((tensao_cisalhante_esc * tensao_cisalhante_media(d)) / (Decimal(tensao_ruptura) * Decimal(0.8)))
        delta2 = b2*b2 - (4*a2*(-c2))
        return (Decimal(-b2) - Decimal(math.sqrt(delta2)) ) / (Decimal(2)*a2)

#critério ASME
elif criterio == 4:
    def tensao_normal(d):
        return math.sqrt((((lim_esc * tensao_normal_var(d))/lim_normal_res)**2) + ((tensao_normal_media(d))**2))

    def tensao_cisalhante(d):
        return math.sqrt((((tensao_cisalhante_esc * tensao_cisalhante_var(d))/lim_cisalhante_res)**2) + ((tensao_cisalhante_media(d))**2))

#critério escoamento ampliado
elif criterio == 5:

```

```
def tensao_normal(d):
    return tensao_normal_var(d) + tensao_normal_media(d)

def tensao_cisalhante(d):
    return tensao_cisalhante_var(d) + tensao_cisalhante_media(d)
```

No início do código mostrado acima, foi criada a variável `parar` sendo igual a `False`. Isso foi feito para ser utilizado na estrutura de repetição `while` na linha seguinte porque os códigos a seguir serão executados até atingir o critério de parada.

```
parar = False

while parar == False:
    if 2.79 <= d0*10**3 <= 51:
        kb = ((d0* 10**3) / Decimal(7.62))**Decimal(-0.107)
    elif 51 < d0*10**3 <= 254:
        kb = Decimal(1.51) * ((d0* 10**3)**Decimal(-0.157))
    print(f'kb: {kb}')
    lim_normal_res = Decimal(ka) * Decimal(kb) * Decimal(kc) * Decimal(kd)
    * Decimal(1) * Decimal(kf) * Decimal(0.5) * Decimal(tensao_ruptura)
    lim_cisalhante_res = Decimal(ka) * Decimal(kb) * Decimal(kc) * Decimal
    (kd) * Decimal(0.577) * Decimal(kf) * Decimal(0.5) * Decimal(tensao_rup
    tura)

    d1 = newton(f,d0,10e-8,20)

    print(f'd0:{d0} ---- d1:{d1}')
    if abs(d1-d0) < 10e-8:
        parar = True
    else:
        d0 = d1

print(d1)
```

Antes de comentá-lo, é preciso falar sobre o valor de k_b , calculado de acordo com a Eq. (4.4), que necessita do valor do diâmetro para encontrá-lo. O problema é que ele precisa do k_b para encontrar a resposta. Para solucionar este empasse, primeiro é calculado o valor dessa variável levando em conta o diâmetro inicial (d_0), encontrado de acordo com a Eq. (5.21). Depois, os limites de resistência normal (σ_n) e cisalhante (τ_n) são feitos e em seguida o método de Newton Raphson é utilizado com os parâmetros f (representa a função utilizada para encontrar o diâmetro), o $d0$ (diâmetro inicial), o ε (tolerância para o erro relativo) e o k (número máximo de iterações).

$$d_0 = \left(\frac{32}{\pi \cdot \sigma_{adm}} \cdot \sqrt{M^2 + \frac{3}{4} T^2} \right)^{\frac{1}{3}} \quad (5.21)$$

Após esse processo, finalmente chega-se no critério de parada que diz: se o valor absoluto da diferença entre o dI , diâmetro calculado pelo método de Newton Raphson, e o $d0$, for menor do que 10^{-8} , valor empírico, a variável parar recebe True, processo finaliza e o valor do diâmetro é encontrado. Caso contrário, representado pelo “else”, a variável $d0$ recebe o dI e o tudo é executado novamente.

Tudo o que foi dito, desde o cálculo da tensão normal máxima até agora, é feito para os exercícios classificados como exercício de projeto. Eles recebem essa classificação porque são feitos para projetar algum elemento de máquina, normalmente o diâmetro (d). Nesses casos, o d é um valor que precisa ser encontrado e o limite de escoamento (σ_e) e o fator de segurança (N) são dados de entrada.

Já nos exercícios de análise, o diâmetro é dado de entrada e o fator de segurança é um dado de saída. Nesse caso o código ficaria da seguinte forma:

```
#cálculo das tensões
tensao_cisalhante_esc = Decimal(0.577) * lim_esc

tensao_normal_max = (kp * ((Decimal(32) * m_max) / (Decimal(pi) *
(d**Decimal(3))))) + (kp * ((Decimal(4) * f_max) / (Decimal(pi) *
(d**Decimal(2)))))
tensao_normal_min = (kp * ((Decimal(32) * m_min) / (Decimal(pi) *
(d**Decimal(3))))) + (kp * ((Decimal(4) * f_min) / (Decimal(pi) *
(d**Decimal(2)))))
tensao_cisalhante_max = (kp * ((Decimal(16) * t_max) / (Decimal(pi) *
(d**Decimal(3)))))
tensao_cisalhante_min = (kp * ((Decimal(16) * t_min) / (Decimal(pi) *
(d**Decimal(3)))))

tensao_normal_var = (tensao_normal_max - tensao_normal_min) / Decimal(2)
tensao_normal_media = (tensao_normal_max + tensao_normal_min) / Decimal(2)

tensao_cisalhante_media = (tensao_cisalhante_max + tensao_cisalhante_min) /
Decimal(2)
tensao_cisalhante_var = (tensao_cisalhante_max - tensao_cisalhante_min) /
Decimal(2)

#cálculo do fator de segurança e escolha dos critérios
if criterio == 1:
    #sorderberg
    def
sorderberg(tensao_normal_media,tensao_normal_var,lim_esc,lim_normal_res,tensao_cisalhante_media,tensao_cisalhante_var,tensao_cisalhante_esc,lim_cisalhante_res):
        tensao_normal_sorderberg = tensao_normal_media + (tensao_normal_var *
(lim_esc / lim_normal_res))
        tensao_cisalhante_sorderberg = tensao_cisalhante_media +
(tensao_cisalhante_var * (tensao_cisalhante_esc / lim_cisalhante_res))
```

```

    fat_seg_sorderberg = lim_esc / ((tensao_normal_sorderberg**Decimal(2))
+ (Decimal(3) *
(tensao_cisalhante_sorderberg**Decimal(2))))**Decimal(1/2))

```

```

    return print(f'Tensão normal sorderberg: {tensao_normal_sorderberg}
\nTensão cisalhante sorderberg: {tensao_cisalhante_sorderberg} \nN
sorderberg: {fat_seg_sorderberg}')

```

```

sorderberg(tensao_normal_media,tensao_normal_var,lim_esc,lim_normal_res,tensao_cisalhante_media,tensao_cisalhante_var,tensao_cisalhante_esc,lim_cisalhante_res)

```

```

elif criterio == 2:

```

```

    #modificado de goodman

```

```

    def

```

```

mod_good(lim_esc,tensao_normal_var,lim_normal_res,tensao_normal_media,tensao_
ruptura,tensao_cisalhante_esc,tensao_cisalhante_var,lim_cisalhante_res,tensao_cisalhante_media):

```

```

    tensao_normal_mod_god = lim_esc * ((tensao_normal_var/lim_normal_res) +
(tensao_normal_media/Decimal(tensao_ruptura)))

```

```

    tensao_cisalhante_mod_god = tensao_cisalhante_esc *
((tensao_cisalhante_var/lim_cisalhante_res) +
(tensao_cisalhante_media/(Decimal(tensao_ruptura)*Decimal(0.8))))

```

```

    fat_seg_mod_god = lim_esc / (((tensao_normal_mod_god**Decimal(2)) +
Decimal(3) * (tensao_cisalhante_mod_god**Decimal(2))))**Decimal(1/2))

```

```

    return print(f'Tensão normal modificado de goodman:
{tensao_normal_mod_god} \nTensão cisalhante modificado de goodman:
{tensao_cisalhante_mod_god} \nN modificado de goodman: {fat_seg_mod_god}')

```

```

mod_good(lim_esc,tensao_normal_var,lim_normal_res,tensao_normal_media,tensao_
ruptura,tensao_cisalhante_esc,tensao_cisalhante_var,lim_cisalhante_res,tensao_cisalhante_media)

```

```

elif criterio == 3:

```

```

    #gerber

```

```

    def

```

```

gerber(lim_esc,tensao_normal_var,lim_normal_res,tensao_normal_media,tensao_
ruptura):

```

```

    a1 = Decimal(1)

```

```

    b1 = (lim_esc * tensao_normal_var) / lim_normal_res

```

```

    c1 = ((lim_esc * tensao_normal_media) /
Decimal(tensao_ruptura))**Decimal(2)

```

```

    delta1 = b1*b1 - (Decimal(4)*a1*(-c1))

```

```

    tensao_normal_gerber2 = (-b1 - Decimal(math.sqrt(delta1)) ) /
(Decimal(2)*a1)

```

```

    print(f'Tensão normal de gerber: {tensao_normal_gerber2}')

```

```

    a2 = Decimal(1)

```

```

    b2 = (tensao_cisalhante_esc * tensao_cisalhante_var) /
lim_cisalhante_res

```

```

    c2 = ((tensao_cisalhante_esc * tensao_cisalhante_media) /
(Decimal(tensao_ruptura) * Decimal(0.8)))**Decimal(2)

```

```

    delta2 = b2*b2 - (Decimal(4)*a2*(-c2))

    tensao_cisalhante_gerber2 = (-b2 - Decimal(math.sqrt(delta2)) ) /
(Decimal(2)*a2)
    print(f'Tensão cisalhante de gerber: {tensao_cisalhante_gerber2}' )

    fat_seg_gerber = lim_esc / (((tensao_normal_gerber2 **Decimal(2)) +
Decimal(3) * (tensao_cisalhante_gerber2 **Decimal(2)))*(Decimal(1/2)))
    return print(f'N gerber: {fat_seg_gerber}')

gerber(lim_esc,tensao_normal_var,lim_normal_res,tensao_normal_media,tensao_
ruptura)

elif criterio == 4:
    #asme
    def
asme(lim_esc,tensao_normal_var,lim_normal_res,tensao_normal_media,tensao_ci
salhante_esc,tensao_cisalhante_var,lim_cisalhante_res,tensao_cisalhante_med
ia):
        tensao_normal_asme = Decimal(math.sqrt((((lim_esc *
tensao_normal_var)/lim_normal_res)**Decimal(2)) +
((tensao_normal_media)**Decimal(2))))
        tensao_cisalhante_asme = Decimal(math.sqrt((((tensao_cisalhante_esc *
tensao_cisalhante_var)/lim_cisalhante_res)**Decimal(2)) +
((tensao_cisalhante_media)**Decimal(2))))
        fat_seg_asme =
lim_esc/Decimal(math.sqrt((tensao_normal_asme**Decimal(2)) +
Decimal(3)*(tensao_cisalhante_asme**Decimal(2))))

        print(f'Tensão normal ASME: {tensao_normal_asme}')
        print(f'Tensão cisalhante ASME: {tensao_cisalhante_asme}')
        print(f'N ASME: {fat_seg_asme}')
        return

asme(lim_esc,tensao_normal_var,lim_normal_res,tensao_normal_media,tensao_ci
salhante_esc,tensao_cisalhante_var,lim_cisalhante_res,tensao_cisalhante_med
ia)

elif criterio == 5:
    #escoamento ampliado
    def
esc_amp(tensao_normal_var,tensao_normal_media,tensao_cisalhante_var,tensao_
cisalhante_media,lim_esc):
        tensao_normal_escamp = tensao_normal_var + tensao_normal_media
        tensao_cisalhante_escamp = tensao_cisalhante_var +
tensao_cisalhante_media
        fat_seg_escamp = lim_esc/((tensao_normal_escamp**Decimal(2)) +
Decimal(3)*(tensao_cisalhante_escamp**Decimal(2)))*(Decimal(1/2))

        print(f'Tensão normal escoamento ampliado: {tensao_normal_escamp}')
        print(f'Tensão cisalhante escoamento ampliado:
{tensao_cisalhante_escamp}')
        print(f'N escoamento ampliado {fat_seg_escamp}')

```

```
return
```

```
esc_amp(tensao_normal_var,tensao_normal_media,tensao_cisalhante_var,tensao_
cisalhante_media,lim_esc)
```

Onde primeiro é feito o cálculo das tensões e depois o critério é escolhido arbitrariamente e o código segue a mesma estrutura de decisão citada anteriormente sendo cada critério com o mesmo número equivalente. No final é dado de saída o fator de segurança, tensão normal e a tensão cisalhante.

6. PARAFUSOS

Os parafusos são frequentemente encontrados em desenhos de engenharia, sua função é unir, não permanentemente, partes de uma estrutura bastando apertá-lo. Podem se diferenciar pelo formato da cabeça, corpo e ponta, onde cada tipo é escolhido de acordo com a função que ele será empregado. Após conhecer a sua função na estrutura, é necessário projetá-lo, encontrar seu diâmetro e calcular as forças aplicadas no elemento.

Nesta seção, dois tipos de exercícios foram automatizados. O primeiro trata-se de uma questão de projeto, em que é necessário encontrar o diâmetro do parafuso de fixação, enquanto o segundo é para cálculo da força e o diâmetro no parafuso através de cargas excêntricas. Antes de começar a trabalhar com as fórmulas, é necessário especificar alguns pontos do parafuso representado na Figura 21.

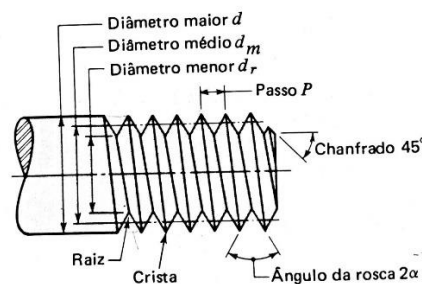


Figura 21: Medidas principais da rosca de um parafuso (SHIGLEY, 1984)

Nessa figura, temos que:

- passo – Distância entre dois pontos correspondentes de filetes adjacentes;
- diâmetro maior ou diâmetro nominal – É o maior diâmetro da parte roscada;
- diâmetro menor ou diâmetro raiz – É o menor diâmetro da parte roscada;
- diâmetro médio – É a medida dos diâmetros maior e menor;
- altura do filete – Metade da diferença entre os diâmetros maior e menor.

Depois de se familiarizar com algumas partes do parafuso, é possível entender os elementos que compõe as fórmulas, que serão introduzidas nos tópicos a seguir.

6.1 DIÂMETRO, DIÂMETRO EFETIVO E ÁREA DE RESISTÊNCIA À TRAÇÃO

De acordo com (SHIGLEY, 1984), a resistência à tração do parafuso é igual a resistência à tração de um cilindro do mesmo material, cujo diâmetro é a média entre o efetivo e o menor (raiz). Esse primeiro pode ser calculado da seguinte forma:

$$d_{ef} = d - 0,649519 \cdot p \quad (6.1)$$

$$d_0 = \frac{d_{ef} + d_r}{2} \quad (6.2)$$

A área de resistência à tração do parafuso é:

$$A_t = \frac{\pi}{4} d_0^2 \quad (6.3)$$

6.2 PRÉ-CARREGAMENTO

Uma junta simplesmente aparafusada é usada quando é necessária uma união desmontável sem a necessidade de destruí-la, por isso ela deve ter boa resistência a fim de aguentar cargas externas simples ou combinadas. Nessa união, o parafuso é apertado com o objetivo de produzir uma carga inicial tracionada F_i , fazendo com que as peças fiquem comprimidas. Após isso, força externa de tração P é aplicada, o parafuso fica mais carregado e as mesmas sofrem alívio de carga.

As forças citadas acima e o esquema da junta está representada na Figura 22 a seguir:

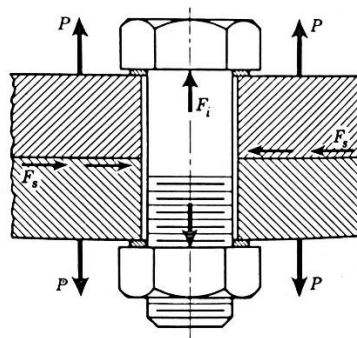


Figura 22: Montagem com pré-carga (SHIGLEY, 1984)

Segundo (SHIGLEY, 1984), esse pré-carregamento tem como efeito aumentar a resistência à carga de tração para as partes de compressão e elevar o atrito entre as peças, dessa forma a resistência à carga de cisalhamento é alta.

Temos, então, as seguintes forças:

- P – Carga externa aplicada à união;
- F_i – Força inicial ou pré-carga;
- F_p – Carga total sobre o parafuso.
- R_p – Resistência de prova

Em que, segundo (BUDYNAS, 2011):

$$F_i = \begin{cases} 0,90R_p \cdot A_t & \text{para uma fixação permanente} \\ 0,75R_p \cdot A_t & \text{para uma fixação não permanente} \end{cases} \quad (6.4)$$

Em situações em que o valor da resistência de prova não é informado, deve-se usar a Eq. (6.5).

$$R_p = 0,855 \cdot \sigma_e \quad (6.5)$$

A carga total sobre o parafuso é dada pela Eq. (6.6):

$$F_p = \frac{k_p}{k_p + k_m} \cdot P + F_i \quad (6.6)$$

Onde k_p é a constante de rigidez do parafuso e k_m é a constante de rigidez da montagem, sendo calculadas a partir das seguintes Equações:

$$k_p = \frac{\pi d^2 \cdot E}{4L} \quad (6.7)$$

$$k_m = \frac{2\pi \cdot d^2 \cdot E}{L} \quad (6.8)$$

Tratando-se de um mesmo material, (SHIGLEY, 1984) recomenda-se usar a relação estabelecida pela Eq. (6.9), usada na automatização dos exercícios, que correlaciona as Eqs. (6.7) e (6.8).

$$k_m = 8k_p \quad (6.9)$$

6.3 CÓDIGO

Os dados de entrada desse código são:

- limite de escoamento (*lim_esc*)
- tensão de ruptura (*tensão_ruptura*)
- carga máxima (*p_max*)
- montagem (*montagem*) podendo ser fixação permanente ou fixação não permanente
- tipo de aço (*aco*)
- fator de segurança (*fat_seg*)
- confiabilidade (*conf*)
- temperatura (*temp*)
- carga (*carga*)
- *kp'* (*kp_linha*)

Após calcular os valores de *k* e determinar o limite de resistência à fadiga que já foi introduzido na seção 4, pode-se construir o código mostrado a seguir.

```
#Cálculo da resistência de prova
rp = lim_esc * 0.855

#Cálculo da Força Inicial
if montagem == 'fixação permanente':
    def fi(a):
        return (0.9 * rp * a)
else:
    def fi(a):
        return (0.75 * rp * a)

def tensao_variavel(a):
    return (1/9 * ( (kp_linha* p_max)/ (2*a)))

def tensao_media(a):
    return tensao_variavel(a) + ((fi(a))/(a))

#Determinação da função utilizada no método de Newton
def f(a):
    return 1 - (((fat_seg * tensao_variavel(a))/(lim_normal_res))**2) -
    (((fat_seg * tensao_media(a))/(lim_esc))**2)

#Cálculo da derivada da função
def flin(f, a):
    if a == 0:
        a = 1
        h = a / 5471
        flin = (1/(12*h)) * ( f(a- 2*h) - 8*f(a - h) + 8*f(a+h) - f(a + 2*h) )
    else:
        h = a / 5471
        flin = (1/(12*h)) * ( f(a- 2*h) - 8*f(a - h) + 8*f(a + h) - f(a + 2*h)
    )
    return flin
```

```

#Cálculo do valor inicial da área
a0 = p_max / lim_esc
print(a0)

#Método de newton
def newton(f, a0, epsilon, maxIter=20):
    if abs(f(a0)) <= epsilon:
        return (a0)

    print("k\t a0\t\t f(a0)") #cabeçalho da tabela que retornará com os
    valores embaixo. \t é o espaçamento que dou para ficar visual.

    k=1
    while k <= maxIter:
        a1 = a0 - f(a0)/flin(f, a0)

        print("%a\t%e\t%e" % (k, a1, f(a1)))

        if abs(f(a1)) <= epsilon:
            return a1

        a0 = a1
        k = k + 1

    print('ERRO: Número máximo de iterações atingido')

    return a1

#Resposta
raiz = newton(f, a0, 10**-9, 20)

print(raiz)

```

Nessa parte do código também se fez necessário o uso do Método de Newton Raphson utilizando como um dos parâmetros o valor inicial da área (a_0) dado pela Eq. (6.10).

$$a_0 = \frac{P}{\sigma_e} \quad (6.10)$$

Com esse resultado de área resistente à tração, podemos consultar a Tabela 3 para encontrar o diâmetro correspondente. Vale ressaltar que durante a automatização dos exercícios em Python, usou-se como parâmetro um código desenvolvido na linguagem Pascal pelo professor orientador. No caso desse exercício, notou-se discrepância nos resultados.

Tabela 3: Diâmetros e áreas de roscas unificadas UNC e UNF (BUDYNAS, 2011)

Designação de tamanho	Diâmetro maior nominal in	Roscas por polegada N	Série grossa – UNC		Roscas por polegada N	Série fina – UNF	
			Área de tensão A_t in ² de tração	Área de diâmetro menor A_r in ²		Área de tensão de tração A_t in ²	Área de diâmetro menor A_r in ²
0	0,600				80	0,001 80	0,001 51
1	0,0730	64	0,002 63	0,002 18	72	0,002 78	0,002 37
2	0,0860	56	0,003 70	0,003 10	64	0,003 94	0,003 39
3	0,0990	48	0,004 87	0,004 06	56	0,005 23	0,004 51
4	0,1120	40	0,006 04	0,004 96	48	0,006 61	0,005 66
5	0,1250	40	0,007 96	0,006 72	44	0,008 80	0,007 16
6	0,1380	32	0,009 09	0,007 45	40	0,010 15	0,008 74
8	0,1640	32	0,014 0	0,011 96	36	0,014 74	0,012 85
10	0,1900	24	0,017 5	0,014 50	32	0,020 0	0,017 5
12	0,2160	24	0,024 2	0,020 6	28	0,025 8	0,022 6
$\frac{1}{4}$	0,2500	20	0,031 8	0,026 9	28	0,036 4	0,032 6
$\frac{5}{16}$	0,3125	18	0,052 4	0,045 4	24	0,058 0	0,052 4
$\frac{3}{8}$	0,3750	16	0,077 5	0,067 8	24	0,087 8	0,080 9
$\frac{7}{16}$	0,4375	14	0,106 3	0,093 3	20	0,118 7	0,109 0
$\frac{1}{2}$	0,5000	13	0,1419	0,125 7	20	0,159 9	0,148 6
$\frac{9}{16}$	0,5625	12	0,182	0,162	18	0,203	0,189
$\frac{5}{8}$	0,6250	11	0,226	0,202	18	0,256	0,240
$\frac{3}{4}$	0,7500	10	0,334	0,302	16	0,373	0,351
$\frac{7}{8}$	0,8750	9	0,462	0,419	14	0,509	0,480
1	1,0000	8	0,606	0,551	12	0,663	0,625
$1\frac{1}{4}$	1,2500	7	0,969	0,890	12	1,073	1,024
$1\frac{1}{2}$	1,5000	6	1,405	1,294	12	1,581	1,521

6.4 CARREGAMENTO EXCÊNTRICO

No exercício de carregamento excêntrico é importante compreender o conceito de sistemas de forças mecanicamente equivalentes, pois todos os exercícios foram trabalhados a partir dele. Nesses sistemas, não ocorre alteração no equilíbrio do corpo se substituirmos as forças aplicadas em um ponto pela sua resultante aplicadas nesse mesmo ponto. Do mesmo modo, o equilíbrio não é alterado ao transportar o ponto de aplicação de uma força ao longo de sua linha de ação. Assim, a resultante e o momento são os mesmos em relação a um polo.

Na questão utilizada para o teste da automação, o sistema foi trabalhado conforme a Figura 23, que mostra do lado esquerdo uma carga excêntrica P aplicada na extremidade livre, cuja linha de ação tem uma distância L em relação ao centroide G dos parafusos e no lado esquerdo, mostra a movimentação da carga para o ponto G , obtendo um sistema mecanicamente equivalente ao primeiro com um momento $M_g = L \cdot P$.

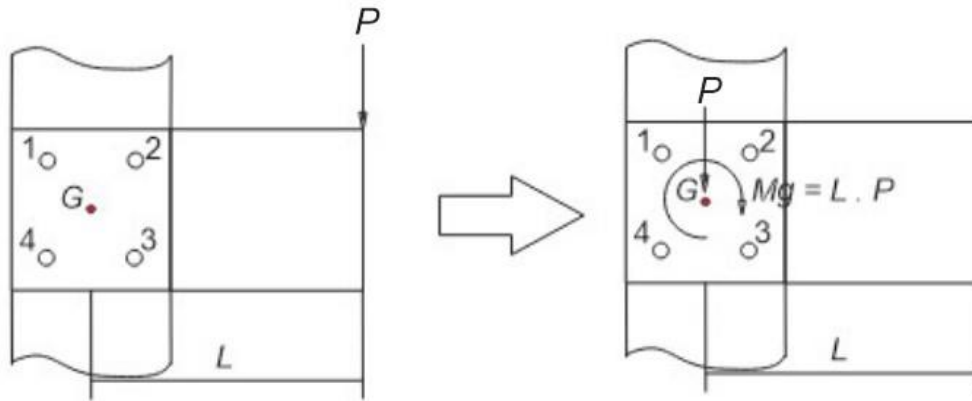


Figura 23: Esquematização de sistemas mecanicamente equivalentes

A partir do sistema esquematizado à direita e pela superposição dos carregamentos apresentados, pode-se obter dois novos sistemas, sendo um deles apenas com uma carga P aplicada no centroide e o segundo formado pelo momento M_g . Com base nisso, a carga aplicada sobre um parafuso pode ser decomposta em duas cargas distintas P_i e Q_i , representadas na Figura 24, onde P_i é a divisão igual da carga P pelo número de parafusos e Q_i é escolhida de forma que seja direcionada perpendicularmente ao segmento de reta formado pela distância do centroide e o parafuso. Tal consideração é feita para todos os elementos de união do sistema.

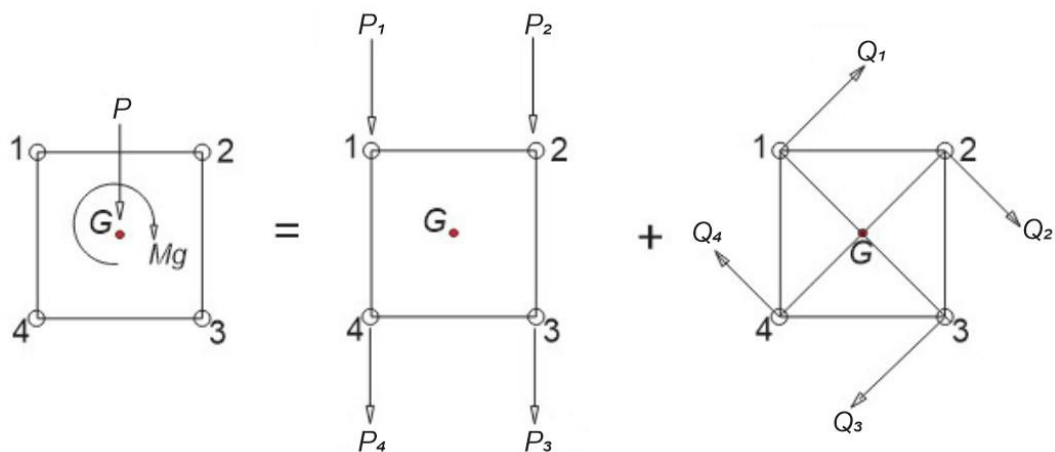


Figura 24: Direção das cargas P_i e Q_i

Segundo (HIGDON, 1981), ao se aplicar uma carga como mostrada na Figura 24, a placa sofre uma rotação elástica em torno do centroide do grupo de parafusos. Hipoteticamente, toda linha reta da placa continua reta após a aplicação do esforço. Assim, as deformações serão maiores nos parafusos mais distanciados do centroide. Por conclusão, estes suportarão maiores

esforços. Portanto, a carga Q_i pode ser encontrada levando em consideração que o parafuso mais distante é o que suporta maior esforço.

6.5 CÓDIGO

Nesta seção faz-se necessário o uso de uma metodologia diferente em função da complexidade do código. Portanto, serão mostradas as fórmulas utilizadas nos exercícios, um resumo teórico e a explicação do código em cada etapa.

Na Figura 25 a seguir, é ilustrada a montagem excêntrica do exercício padrão utilizado para automação.

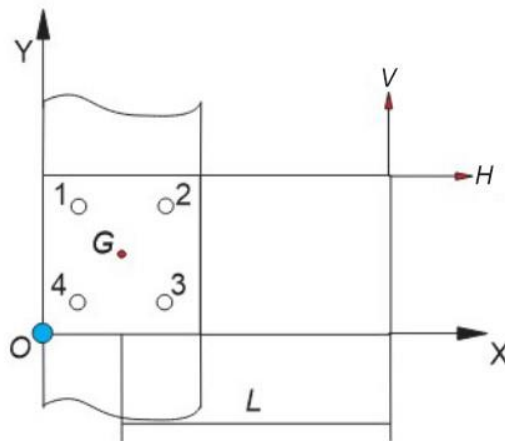


Figura 25: Representação da montagem excêntrica utilizada no exercício automatizado desta seção

Os dados de entrada do código desse exercício são:

- lista de listas (f) que representam as cargas excêntricas aplicadas em seus devidos eixos
- ponto de aplicação das cargas (F), com uma lista cujo índice 0 representa o eixo x e o 1 o y .
- posição dos parafusos (p)
- área (a) dos parafusos
- diâmetro (d) dos parafusos
- espessura da chapa mais fina (t)

```
# valor das cargas aplicadas
f = [[32000, -16000]]

#ponto de aplicação das cargas
F = [0.500, 0.160]

#posicionamento dos parafusos
p = [[0, 0.120], [0.150, 0.120], [0.150, 0], [0, 0]]
```

```
#áreas dos parafusos
a = [[1], [1], [1], [1]]

t = Decimal(0.01)
d = Decimal(0.016)
```

Inicialmente, deve-se arbitrar um sistema de referência XYZ, indicando a origem O do sistema e numerar os parafusos de forma que o parafuso número 1 não fique na origem. A seguir, informa-se a posição (x_i, y_i) de cada parafuso em relação ao sistema, a partir disso, pode-se encontrar a posição $G(\bar{x}, \bar{y})$ do centroide através das fórmulas a seguir, onde A_i representa a área da seção transversal do furo onde o parafuso i está inserido.

$$\bar{x} = \frac{\sum_{i=1}^n x_i \cdot A_i}{\sum_{i=1}^n A_i} \quad (6.11)$$

$$\bar{y} = \frac{\sum_{i=1}^n y_i \cdot A_i}{\sum_{i=1}^n A_i} \quad (6.12)$$

A seguir, está demonstrada a parte do código responsável pelo cálculo do centroide.

```
#Cálculo do centroide
i = 0
b = []
c = []

for i in range(0, len(p)):
    x_barra = (p[i][0] * a[i][0]) / len(p) * a[i][0]
    y_barra = (p[i][1] * a[i][0]) / len(p) * a[i][0]

    x_barra = b.append(x_barra)
    y_barra = c.append(y_barra)

x_barra = sum(b)
y_barra = sum(c)

i = i + 1

G = [x_barra, y_barra]
```

Nota-se o uso do contador i , que recebe o valor 0 no início porque essa etapa será executada a quantidade de vezes equivalente ao número de parafusos informados como dado de entrada, para essa repetição isso usa-se a estrutura “for”. Ao final de cada etapa, os resultados serão armazenados nas listas b e c , que inicialmente estavam vazias.

Nessa parte do código serão feitos os cálculos para encontrar o $G(\bar{x}, \bar{y})$ ($G = [x_barra, y_barra]$). Primeiro é feito o cálculo do x_barra , utilizando o mesmo princípio representado pela Eq. 6.10. Contudo, ele é feito em partes, esquematizadas a seguir:

- Realização do cálculo de $\frac{x_i \cdot A_i}{A_i}$, onde i corresponde ao número do parafuso
- Armazenamento do valor na lista b .
- O mesmo processo é feito paralelamente para encontrar o y_barra .
- Ao final, todos os valores armazenados na lista são somados, se tornando igual as Eqs. 6.11 e 6.12.

Para dar seguimento ao exercício, precisamos fornecer o ponto A de aplicação da carga para encontrar o vetor GA ($G - A$). A fim de evitar a repetição da variável GA no código, o que aumentaria a possibilidade de erro, optou-se pelo uso da variável GF , a distância do ponto A em relação ao centroide, para o cálculo do momento resultante

```
#Distância entre o centroide e o ponto de aplicação das cargas (GF)
GF = [F[0]-G[0], F[1]-G[1]]

#Vetor GA
i = 0
x = []

while i < len(p):
    GA = [p[i][0] - G[0], p[i][1]-G[1]]
    x.append(GA)
    i = i + 1

GA = x
```

A distância entre o centroide e o ponto de aplicação das cargas foi calculado através da subtração ordenada entre os elementos da lista F e G .

O vetor GA foi feito de forma parecida com o cálculo do centroide, todavia foi utilizada apenas a lista x , inicialmente vazia, e a subtração foi feita pela linha $GA = [p[i][0] - G[0], p[i][1]-G[1]]$.

Como o objetivo do código é deixar o exercício o mais genérico possível, para ser utilizado em outros casos, foi adicionado o contador i e utilizado a estrutura de repetição “while”, pois assim, o código será repetido até o tamanho da lista p , ou seja, o número de parafusos do sistema.

Para o cálculo do primeiro elemento de GA , faz-se a subtração entre o primeiro elemento da lista utilizada no primeiro índice das lista de p ($p[0][1]$) e o primeiro índice da lista G ($G[0]$).

A subtração entre o segundo elemento da lista utilizada no primeiro índice das lista de p ($p[0][1]$) e o segundo índice da lista G ($G[1]$) foi empregada para fornecer o segundo elemento da lista x . Isso é feito para os elementos correspondentes a ordenada do sistema de referências. Esse mesmo processo é feito para os outros elementos da lista p . Após esse processamento, a variável GA recebe os valores da lista x .

Em seguida, calcula-se as distâncias r_i entre o centroide G e o parafuso i .

$$r_i = \sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2} \quad (6.12)$$

```
i = 0
r = []

while i < len(p):
    x = math.sqrt((p[i][0] - G[0])**2 + (p[i][1] - G[1])**2)
    r.append(x)

    i = i+1
```

O código acima representa a Eq. (6.12), seguindo os mesmos princípios utilizados nos códigos do cálculo do centroide e do vetor GA .

O momento resultante M_G , no código representado como MR , e seu módulo são determinados a partir das seguintes Equações:

$$M_G = GA \times V + GA \times H \quad (6.14)$$

$$M_G = |M_G| \quad (6.15)$$

```
j = 0
MR = []
while j < len(f):
    y = prod_vet([GF], [f[j]])
    MR.append(y)
    j = j + 1

MR = sum(MR)
```

Para calcular o momento resultante, foi necessário o uso da função “prod_vet”, a mesma utilizada para automatizar o exercício de projeto da seção 5 da apostila. No final do código, foi feita a soma do primeiro elemento, que corresponde a $GA \times V$ com o segundo elemento, $GA \times H$, de MR, semelhante ao realizado na Eq. (6.14).

Para encontrar a força Q , segundo (SHIGLEY, 1984), usa-se a Eq. (6.16), entretanto é necessário dividir previamente igualmente as cargas V e H entre os parafusos.

$$Q_i = \frac{r_i M_g}{r_1^2 + r_2^2 + \dots + r_n^2} \quad (6.16)$$

No código, antes de calcular a força Q , foi necessário encontrar o seu vetor. Foi utilizada a estrutura de decisão “if” para determinar a direção do momento. De modo que, se o valor de MR for positivo, as forças Q s apontam para o sentido anti-horário, se for negativo, eles apontam para o sentido horário. Depois de calculado o vetor, é preciso encontrar sua direção através do vetor unitário (u).

```
# vetor Q
if MR > 0:
    i = 0
    while i < len(GA):
        GA[i][0], GA[i][1] = -GA[i][1], GA[i][0]
        i = i + 1
else:
    i = 0
    while i < len(GA):
        GA[i][0], GA[i][1] = GA[i][1], -GA[i][0]
        i = i + 1
Q = GA

#cálculo do vetor unitário para saber a direção de Q
i = 0
x = []
while i < len(Q):
    try:
        u = [Q[i][0]/math.sqrt( ((Q[i][0])**2) + ((Q[i][1])**2)
), Q[i][1]/math.sqrt( ((Q[i][0])**2) + ((Q[i][1])**2) )]
        x.append(u)
    except:
        u = [0.0, 0.0]
        x.append(u)
        i = i + 1

u = x
```

Foi utilizada a estrutura “try/except” no código do vetor unitário, pois, em alguns exercícios, o programa estava acusando divisão por zero e retornava erro. Isso acontece quando um parafuso está no mesmo ponto que o centroide, dessa forma não há vetores perpendiculares. Sendo assim, quando o evento citado ocorre, o código retorna o valor [0.0 , 0.0] e adiciona na lista auxiliar criada anteriormente. Quando não, a conta é feita normalmente.

Uma vez que foi encontrado o vetor e seu unitário, agora sim pode-se calcular o valor de Q conforme representado na Eq. (6.16).

```
#tratamento de dados para elevar a segunda potência todos os elementos da
matriz r e depois somá-los.
i = 0
x = []
while i < len(r):
    a = r[i]**2
    x.append(a)
    i = i+1
a = sum(x)
#Cálculo da força Q
i = 0
q = []

while i < len(r):
    Q = r[i]*MR / a
    q.append(Q)
    i = i + 1
```

Foi necessário elevar todos os elementos de r ao quadrado, por causa da Eq. (6.16), esses valores foram somados e adicionados na lista a . A partir disso, pode-se calcular o valor de Q propriamente dito. Ao final, tem-se uma lista q com todas as forças definidas. Completando essa parte, multiplica-se todos os valores de q pelos de u .

Por conta disso, as linhas de código foram feitas da seguinte forma:

```
#multiplicando a força Q pelo vetor unitário
i = 0
x = []

while i < len(q):
    Q = [q[i]*u[i][0], q[i]*u[i][1]]
    x.append(Q)
    i = i + 1
Q = x
```

Ao final tem-se uma lista Q com um elemento correspondendo a um parafuso, dentro deles uma força na direção x e outra na y . Todos esses elementos correspondem aos vetores Q s. Eles serão somados aos valores das forças V , representada por $f[0][0]$, e será dividido, assim como H , representada por $f[0][1]$, pelo número de parafusos ($\text{len}[p]$). Dessa forma são encontrados os vetores F que, no código está representado pela lista x .

```
#encontrando os vetores F
i = 0
x = []

while i < len(p):
    F = [Q[i][0] + (f[0][0]/len(p)), Q[i][1] + (f[0][1]/len(p))]
    x.append(F)
    i = i + 1
```

Em sequência é calculado o módulo de F para que tenhamos todos os valores das forças resultantes em cada parafuso. Elas são causadas pela aplicação das cargas excêntricas V e H citadas no início do exercício. No código esse processo é feito da seguinte forma:

```
i = 0
y = []
while i < len(Q):
    F = [math.sqrt(x[i][0]**2 + x[i][1]**2)]
    y.append(F)
    i = i + 1
F = y

Fi = max(F)
```

Logo abaixo do laço de repetição a variável F recebe y , que corresponde a lista com todos os valores das forças resultantes. Ao final, é criado Fi composto pelo maior valor de F . Isso é necessário para analisar a tensão de cisalhamento (τ_i) e a tensão de compressão superficial (σ_{ci}), onde é utilizada a maior força resultante dos parafusos para calculá-las.

Essas tensões são encontradas através das Equações (6.16), (6.17) e (6.18) demonstradas a seguir:

$$\tau_i = \frac{4F_i}{\pi d^2} \quad (6.16)$$

$$\sigma_{ci} = \frac{F_i}{A_p} \quad (6.17)$$

$$A_p = t \cdot d \quad (6.18)$$

Representadas no código, quando o exercício é de análise, como:

```
tensao_cisalhamento = (4*Fi[0]) / (3.14* d**2)
tensao_compressao = Fi[0] / (d*t)
```

Já, quando é de projeto, é necessário encontrar o valor do diâmetro do parafuso:

```
d1 = math.sqrt((4*Fi)/(Decimal(3.14)*tensao_cis))
d2 = Fi/(t*tensao_compressao)

if d1>d2:
    print(d1)
else:
    print(d2)
```

São levados em consideração os dois tipos de tensões atuantes no parafuso e, no final, é escolhido aquele que retornar o maior valor.

7. EIXOS E ÁRVORES

Eixos e árvores são elementos rotativos ou estacionários sujeito a cargas variáveis que os levam a sofrer efeitos de fadiga. Por conta disso, esses elementos podem ser dimensionados como qualquer peça sujeita a esforços combinados.

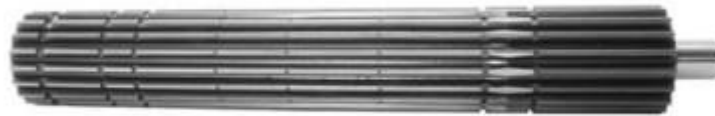


Figura 26: Ilustração de uma árvore principal (REAL TRATOR PEÇAS, 2022)

Esse dimensionamento pode ser feito utilizando o critério da energia de distorção máxima apresentado a algumas seções atrás, onde uma peça estará em segurança se for dimensionada utilizando a Eq. (7.1) a seguir:

$$(\sigma^2 + 3\tau^2)^{\frac{1}{2}} = \sigma_{adm} \quad (7.1)$$

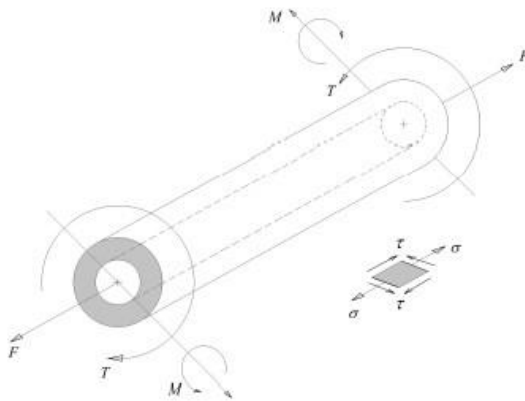


Figura 27: Esforços combinados em uma árvore (LA PLATA, 2022)

Substituindo a tensão normal induzida pela fórmula da combinação de carga axial com flexão e a tensão de cisalhamento pela do momento de torção (ambas já citadas neste trabalho), temos então a Eq. (7.2) utilizada para encontrar o diâmetro no exercício de projeto. Junto a ela

também são adicionados os coeficientes k_m , k_t e α . Segundo (ALBUQUERQUE, 1980), eles são introduzidos para evitar fraturas por fadiga, flambagem ou sobrecarga.

De acordo com (LA PLATA, 2022), usa-se a Eq. (7.2), encontra-se o diâmetro externo (d_0):

$$d_0^3 = \frac{32}{\pi \cdot \sigma_{adm} \cdot (1-k^4)} \left\{ \left[k_m \cdot M + \frac{1}{8} \cdot \alpha \cdot F \cdot d_0 \cdot (1+k^2) \right]^2 + \frac{3}{4} (k_t \cdot T)^2 \right\}^{\frac{1}{2}} \quad (7.2)$$

O diâmetro interno é dado por:

$$d_i = k \cdot d_0 \quad (7.3)$$

$$k = \frac{d_i}{d_0} \quad (7.4)$$

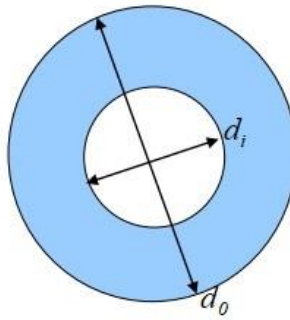


Figura 28: Ilustração do diâmetro interno (d_i) e externo (d_0) de uma árvore (ANDRADE, 2022)

Onde a razão entre os diâmetros (k) é informada no enunciado da questão. Os valores de k_m e k_t podem ser obtidos de acordo com a Tabela 4:

Tabela 4: Valores dos fatores de carga para eixos e árvores (ALBUQUERQUE, 1980)

Tipo de Carga	k_m	k_t
Esforços estáveis e gradualmente aplicados	1,5	1,0
Esforços bruscamente aplicados, com choques moderados	1,5 – 2,0	1,0 – 1,5
Esforços bruscamente aplicados, com choques fortes	2,0 – 3,0	1,5 – 3,0

O fator do fenômeno de flambagem, (α) é determinado, segundo (ALBUQUERQUE, 1980), por:

$$\alpha = \frac{1}{1 - 0,0044 \left(\frac{L}{\rho} \right)^2} \quad (7.5)$$

Sendo a razão $\frac{L}{\rho} < 115$ utilizada neste trabalho.

O termo ρ indica o raio de giro da seção transversal dado por:

$$\rho = \frac{1}{4} \cdot d_0 \cdot \sqrt{1 + k^2} \quad (7.6)$$

Para a tensão admissível (σ_{adm}), de acordo com (HALL, 1981), foi adotado:

$$\sigma_{adm} = \begin{cases} 0,225 \cdot \sigma_e \\ 0,135 \cdot \sigma_e \end{cases} \text{ com rasgo de chaveta} \quad (7.7)$$

$$\sigma_{adm} = \begin{cases} 0,30 \cdot \sigma_e \\ 0,18 \cdot \sigma_e \end{cases} \text{ sem rasgo de chaveta} \quad (7.8)$$

Por fim, deve-se calcular a deflexão angular representada pela Eq. (7.9). De acordo com (HALL, 1981), o valor máximo permitido por comprimento não deve ultrapassar 2,6°/m.

$$\frac{\theta}{L} = \frac{583,61 \cdot T}{G \cdot d_0^4 (1 - k^4)} \text{ graus} \quad (7.9)$$

Onde:

- T é o momento de torção;
- G é o módulo de elasticidade transversal;
- L é o comprimento da árvore.

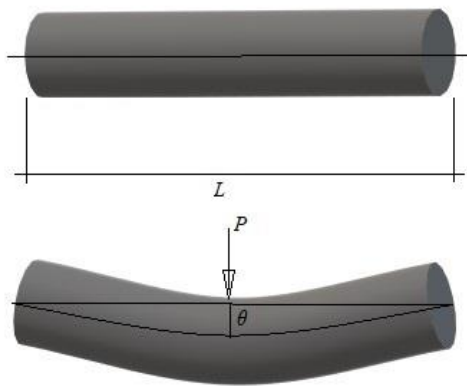


Figura 29: Demonstração de um eixo de comprimento L , sofrendo uma deflexão angular (θ) causada por um carregamento P . O primeiro está sem deformação. O segundo está sofrendo uma deflexão devido a força aplicada.

Para se fazer uma estimativa inicial do diâmetro, pode-se utilizar a Eq. (7.10):

$$d_0^3 = \frac{32}{\pi \cdot \sigma_{adm} \cdot (1 - k^4)} \left[M^2 + \frac{3}{4} \cdot T^2 \right]^{\frac{1}{2}} \quad (7.10)$$

7.1 CÓDIGO

Neste código, tem-se como dados de entrada:

- limite de escoamento (*lim_esc*)
- limite de ruptura (*lim_ruptura*)
- potência (*potencia*), dada em HP deve ser multiplicada por 745,7 para ser convertida em Watts (*pot*)
- número de rotações por minuto (*n*), deve ser multiplicado por 0,1047 para obter o valor em rad/s (*N*)
- momento máximo (*momento_max*)
- força axial (*forca_axial*).

- rasgo de chaveta onde foi criada uma estrutura de decisão com a possibilidade de resposta sim ou não. No primeiro caso a Eq. (7.6) é utilizada e a tensão admissível assume o menor valor possível. No segundo, a Eq. (7.7).
- O comprimento (L)
- relação entre os diâmetros interno e externo (k)
- o tipo de choque (*tipo_choque*) podendo ser moderado, gradual ou forte
- k_m e k_t , disponíveis na Tabela 4.

Esses dados são utilizados para encontrar o valor do diâmetro externo no exercício de projeto do capítulo de Eixos e Árvores. Nele foi necessário utilizar o Método de Newton Raphson, já discutido anteriormente.

```
#determinação da potência, das rotações em rad/s e torque máximo
pot = potencia * 745.7
N = n * 0.1047
torque_max = round(pot / N ,2)

#determinação do km e kt
if tipo_choque == 'moderado':
    km = 2
    kt = 1.5
elif tipo_choque == 'gradual':
    km = 1.5
    kt = 1
elif tipo_choque == 'forte':
    km = 3
    kt = 3

#determinação da tensão admissível
if rasgo_chaveta == 'sim':
    x = 0.225 * lim_esc
    y = 0.135 * lim_ruptura
    if x < y:
        tensao_adm = x
    else:
        tensao_adm = y
else:
    x = 0.3 * lim_esc
    y = 0.18 * lim_ruptura
    if x < y:
        tensao_adm = x
    else:
        tensao_adm = y

#valor inicial do diâmetro externo
d0 = ((32/(3.14*tensao_adm*(1-(k**4))))*(momento_max**2 +
(3/4*(torque_max**2)))**(1/2))**(1/3)

#fórmula utilizada para encontrar o diâmetro no método de newton
def f(d):
    return d**3 - (((32/(3.14*tensao_adm*(1-(k**4))))*(km*momento_max + (1/8
* (1 / (1 - 0.0044*(L/((1/4 * d * math.sqrt(1+ 0.6**2)))))) * forca_axial *
d * (1+ k**2)))**2 + (3/4*(kt*torque_max**2)))**(1/2))
```

```

#encontrar o valor da derivada
def flin(f, d):
    if d == 0:
        d = 1
        h = d / 5471
        flin = (1/(12*h)) * ( f(d- 2*h) - 8*f(d - h) + 8*f(d+h) - f(d + 2*h) )
    else:
        h = d / 5471
        flin = (1/(12*h)) * ( f(d- 2*h) - 8*f(d - h) + 8*f(d+h) - f(d + 2*h) )
    return flin

#determinando o diâmetro utilizando o método de newton
d = newton(f, d0, 10**-8, maxIter=20)

#diâmetro interno
d_int = k*d*10**3

#deflexão angular
def_ang = (583.61 * torque_max) / ((80*10**9) * (d**4) * (1 - (k**4)))

```

Os dados de saída do código acima são:

- diâmetro externo (d)
- diâmetro interno (d_{int})
- deflexão angular, onde o torque máximo ($torque_max$) é encontrado dividindo a potência (pot) pelo número de rotações em rad/s (N).

8. CHAVETAS E ESTRIAS

Segundo (SHIGLEY, 1984), as chavetas são usadas para evitar alguns movimentos, como o axial relativo, entre árvores e elementos como engrenagens. Podem ser retangulares, redondas ou estriadas, essa última, em comparação com as outras, apresenta elevada capacidade de transmissão de momento, pois é como se fossem várias chavetas trabalhando em conjunto.

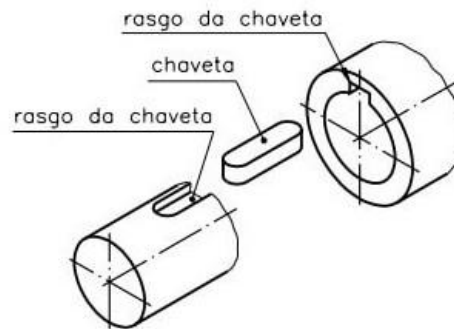


Figura 30: Representação da chaveta em um sistema com eixo e um outro componente (Chaveta mecânica: veja para que serve, principais tipos e aplicações., 2022)



Figura 31: Ligação estriada entre eixo e bucha (Eixos e buchas estriadas, 2022)

O código de automação do exercício desta seção foi feito baseado nas chavetas estriadas, sendo necessário mostrar as equações utilizadas para se chegar ao resultado.

Inicialmente, é necessário converter as unidades da potência (P) e da rotação (ω). A primeira foi dada em HP e mudará para Watts multiplicando o valor dado por 745,7 e a segunda, dada em rpm, se converterá para radianos por segundo ao ser multiplicada por $\frac{2\pi}{60}$.

Em sequência, é calculado o torque (T) segundo a Eq. (8.1) a seguir:

$$T = \frac{P}{\omega} \quad (8.1)$$

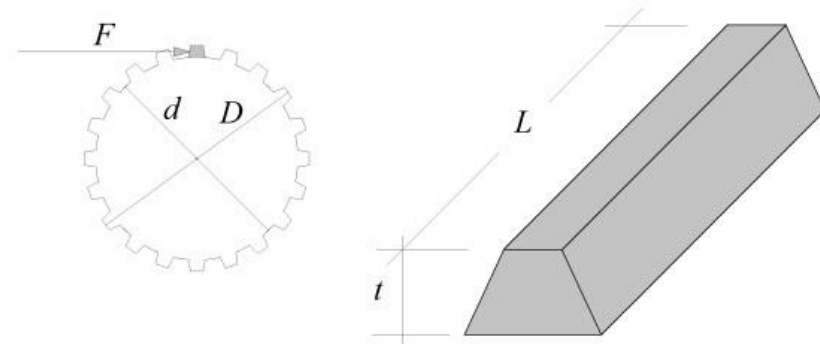


Figura 32: Representação esquemática de uma estria com uma força F aplicada (LA PLATA, 2022)

Agora é possível iniciar os cálculos para projetar a chaveta de fato e encontrar o valor do diâmetro maior (D) e comprimento (L). Esta última variável é encontrada, segundo a (FAIRES, 1976), utilizando a relação com o diâmetro da árvore a seguir:

$$1,25 d \leq L \leq 2d \quad (8.2)$$

O diâmetro maior:

$$D = d + (2 \cdot t) \quad (8.3)$$

8.1 CÓDIGO

Os dados de entrada são:

- diâmetro do eixo (d)
- potência (P)
- pressão admissível (p_{adm})
- número de estrias (n)
- altura (t)
- número de rotações por minuto (rpm)

```
#conversão da potencia em watts e rotação em rad/s
pot = P * 745.7
rot = rpm * 0.1047

#cálculo da torção
T = pot / rot

#cálculo do comprimento
L = d*Decimal(1.25)
```

```
#cálculo do D (diâmetro maior)  
D = d + (Decimal(2)*t)
```

Depois de convertidos, a potência passa a ser representada por “*pot*” e as rotações em rad/s, “*rot*”.

Através desse código consegue-se achar o diâmetro maior (D) e o comprimento (L) da estria.

9. PROJETO E LUBRIFICAÇÃO DE MANCAIS DE DESLIZAMENTO

A função de um mancal é dar suporte aos eixos rotativos ou oscilatório e, também às árvores. A Figura 33, a seguir, representa um tipo de mancal: o de deslizamento, que foi o principal foco neste capítulo da apostila.

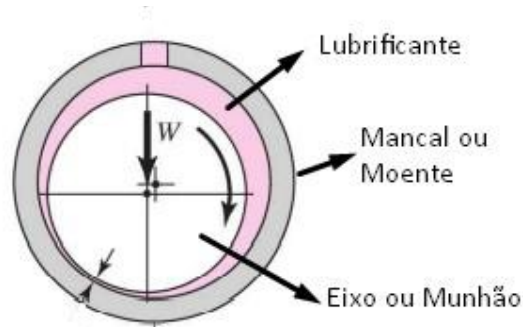


Figura 33: Mancal de deslizamento (BUDYNAS, 2011)

Este tipo, segundo (LA PLATA, 2022), tem como característica os efeitos de atrito mútuo sofridos pelo munhão e moente quando ao dar partida na máquina. Isso acontece porque, quando o equipamento está em repouso, eles se tocam. Por conta disso há uma camada de lubrificante na região de contato dos dois elementos.

À medida que eixo acelera, há efeito de bombeamento do óleo lubrificante que faz esta parte do mancal flutuar. A partir daí não há mais contato entre os elementos, por isso recomenda-se que não haja frequentemente partidas e paradas em máquinas que possuam mancais de deslizamento, evitando o desgaste.

A partir de agora serão introduzidas as fórmulas que estão presentes na apostila e que foram utilizadas no código para encontrar o valor da viscosidade dinâmica (μ) e a temperatura (T).

9.1 MASSA ESPECÍFICA E VISCOSIDADE DO LUBRIFICANTE

A massa específica (ρ) pode ser encontrada, de acordo com (HALL, 1981), de forma empírica em função da sua temperatura (T_0) através fórmula apresentada a seguir, em ρ_{60° é a massa específica do óleo a 60°C .

$$\rho = \rho_{60^\circ} - 0,365 \cdot (T_0 - 60) \quad (9.1)$$

Já para a viscosidade dinâmica, (BUDYNAS, 2011) recomenda utilizar:

$$\mu = \rho \cdot \left(0,22 \cdot t - \frac{180}{t}\right) \times 10^{-6} \quad (9.2)$$

Onde t é a viscosidade de Saybolt em segundos.

9.2 COEFICIENTE DE ATRITO

O coeficiente de atrito para mancais, com base em (HALL, 1981), é dado por:

$$f = 0,334 \cdot \frac{\mu \cdot rpm}{p} \cdot \frac{D}{C} + k \quad (9.3)$$

Onde:

- μ é a viscosidade dinâmica em Pa.s
- D é o diâmetro do eixo
- C a folga diametral
- k é o parâmetro tabelado que depende da razão $\frac{L}{D}$ de acordo com a Figura 34.
- p é a pressão sobre o mancal com base na área projetada calculada, de acordo com (SHIGLEY, 1984), pela Eq. (9.4) mostrada a seguir. Onde L é o comprimento do mancal mostrado na Figura 35. W é a carga radial sobre o mancal, demonstrado na Figura 36.

$$p = \frac{W}{L \cdot D} \quad (9.4)$$

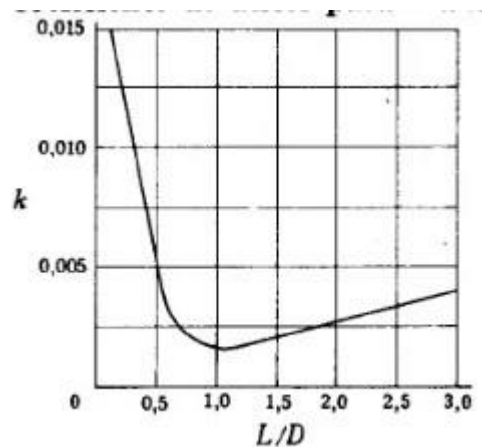


Figura 34: Gráfico de k em função da razão L/D (HALL, 1981)

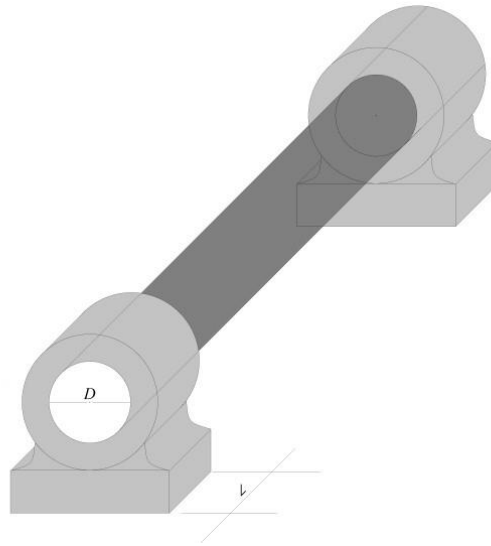


Figura 35: Ilustração do diâmetro e comprimento do mancal de deslizamento (LA PLATA, 2022)

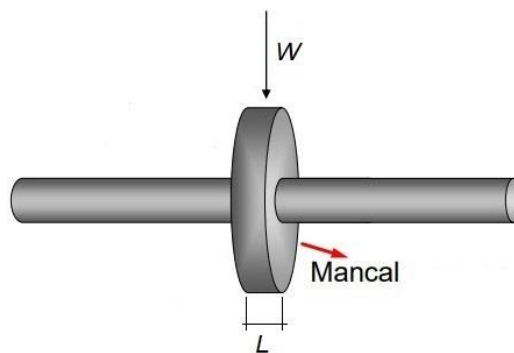


Figura 36: Carga radial aplicada sobre o mancal. Figura adaptada de (ANDRADE, 2022)

9.3 QUANTIDADE DE CALOR GERADA POR ATRITO

Na apostila da disciplina Elementos de Máquinas I é possível encontrar toda a demonstração matemática usada para calcular a quantidade de calor gerada por atrito (Q_g). Como o foco deste trabalho é a automatização dos exercícios, será mostrada apenas a equação a seguir:

$$Q_g = \frac{f \cdot W \cdot \pi \cdot D \cdot rpm}{60} \quad (9.5)$$

9.4 QUANTIDADE DE CALOR DISSIPADA

Segundo (HALL, 1981), a quantidade de calor dissipada (Q_d) de um mancal pode ser determinada pela equação de Lashe a seguir:

$$Q_d = \frac{113,015}{k} (\Delta T + 18,3)^2 L \cdot D \quad (9.6)$$

Onde ΔT é a diferença entre a temperatura do mancal (T) e a temperatura ambiente (T_A).

$$\Delta T = T - T_A \quad (9.7)$$

O valor de k é 31 para mancais pesados com ventilação satisfatória e 55 para mancais leves em ambientes sem ventilação.

9.5 TEMPERATURA DE OPERAÇÃO DO ÓLEO LUBRIFICANTE

A temperatura de operação do óleo lubrificante (T_0), de acordo com (HALL, 1981), pode ser encontrada pela equação a seguir:

$$T_0 = T_A + 2\Delta T \quad (9.8)$$

9.6 NÚMERO DE SOMMERFIELD

O número de Sommerfield (S) é um parâmetro adimensional utilizado nos mancais de deslizamento para se obter, segundo (LA PLATA, 2022), de forma empírica, quantidades como: coeficiente de atrito (f), pressão máxima do óleo do mancal, elevação de temperatura do óleo (ΔT), espessura mínima da película de óleo, entre outros. Representado pela Eq. (9.9) de acordo com (FAIRES, 1976), em que N' é o número de rotações por segundo do eixo.

$$S = \frac{\mu \cdot N'}{p} \cdot \left(\frac{D}{C}\right)^2 \quad (9.9)$$

9.7 CÓDIGO

Os dados de entrada deste código são D , L , C , W , rpm , T , T_a , k e ventilação com as opções de resposta “sim” ou “não” para determinar o valor de K . Todas as variáveis foram representadas com as mesmas letras no código.

```
# encontrar o valor de K
if ventilacao == 'sim':
    K = 31
else:
```

```

K = 55

delta_t = T - Ta

# quantidade de calor dissipada
Qd = (113.015/K) * ((delta_t + 18.3)**2) * L * D

#coeficiente de atrito
#fazendo Qg = Qd
f = (60 * Qd) / (W * 3.14 * D * (rpm))

#pressão sobre o mancal
x = L * D
p = W / x

#viscosidade dinâmica
u = p * ((f - k) / (0.334 * (rpm) * (D/C)) )

#temperatura do óleo lubrificante
To = Ta + (2*delta_t)

#número de sommerfield
S = ((u * (rpm/60)) / p) * (D/C)**2

#massa específica do lubrificante
ro = 900 - (0.365 * (To - 60))

```

Os dados de saída do código são:

- viscosidade dinâmica (u)
- temperatura do óleo (To) lubrificante
- número de Sommerfield (S)
- massa específica do lubrificante (ro)

10. MANCAIS DE ROLAMENTO

Como falado na seção anterior, de acordo com (LA PLATA, 2022), mancais têm a função de dar suporte a árvores ou eixos rotativos. Diferente do mancal de deslizamento, o de rolamento não sofre efeito do atrito. Isso porque neste tipo, o eixo é envolvido por uma peça chamada anel interno que é responsável por sofrer todo o desgaste da peça.

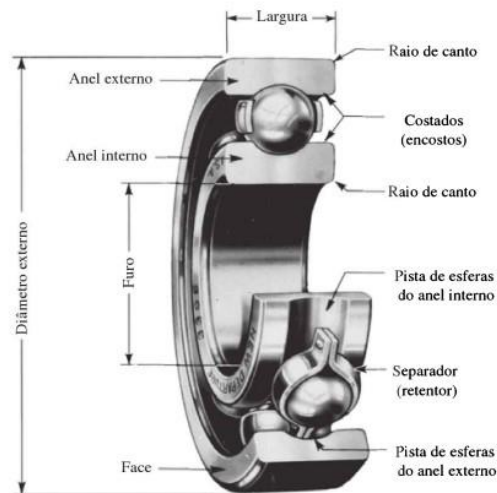


Figura 37: Mancal de rolamento (BUDYNAS, 2011)

Essa deterioração do elemento é causada pelas tensões nele aplicadas e que podem gerar falhas, ou seja, o descascamento da superfície que recebe a carga. Isso ocorre porque essas tensões ultrapassam o limite de resistência à fadiga do material, sendo esperado, portanto, uma vida limitada do rolamento.

O cálculo dessa vida será mostrado na seção a seguir e, junto com ele, a confiabilidade desta estrutura.

10.1 VIDA E CONFIABILIDADE DE UM GRUPO DE ROLAMENTOS

De acordo com (LA PLATA, 2022), a vida de um rolamento (L) pode ser definida como o número total de revoluções ou de horas de operação do rolamento a uma velocidade constante necessária para desenvolver a falha. Já sua vida nominal (L_{10}) é a quantidade de revoluções ou de horas a uma determinada velocidade constante em que 90% de um grupo de rolamentos completará ou excederá para ocorrer a falha.

As Eqs (10.1) e (10.2) são resultantes das distribuições de Weibull, uma distribuição de probabilidade contínua.

$$L_{10} = \frac{L}{6,84 \left[\ln \frac{1}{R} \right]^{0,854}} \quad (10.1)$$

A confiabilidade (R) é a probabilidade de exercerem suas funções de forma adequada nas condições para os quais foram projetados.

$$R(L) = \exp \left[- \left(\frac{L}{6,84 \cdot L_{10}} \right)^{1,17} \right] \quad (10.2)$$

Para entender como as equações acima foram feitas, é necessário consultar o (LA PLATA, 2022).

10.2 CÓDIGO

Os dados de entrada desta seção são a vida do rolamento (L) e sua confiabilidade (R).

```
#cálculo da vida nominal
L10 = L / (6.84 * (log(1/R))**0.854)
```

O dado de saída será a vida nominal (L_{10}).

10.3 SELEÇÃO DE UM ROLAMENTO

Para fazer a seleção de um rolamento, é necessário conhecer alguns pontos, elucidados a seguir.

10.3.1 CARGA DINÂMICA EQUIVALENTE

A carga dinâmica equivalente é definida como uma carga hipotética podendo atuar de maneira radial, axial e central dependendo do tipo de rolamento. Seu cálculo, de acordo com (SHIGLEY, 1984), está representado na equação a seguir:

$$F = X \cdot F_r + Y \cdot F_a \quad (10.3)$$

Onde:

- F_r é a carga radial
- F_a a axial
- X e Y são constantes e em rolamentos de esferas costuma-se adotar $X = 1$ e $Y = 0$.

10.3.2 CAPACIDADE DE CARGA DINÂMICA

A capacidade de carga dinâmica (C) é a carga que um grupo de rolamento pode suportar para uma vida nominal de um milhão de revoluções. Para calcular, de acordo com (BUDYNAS, 2011), é necessário utilizar a fórmula a seguir e considerar $a = 3$ quando for mancais de esfera e $a = 10/3$ para os demais tipos.

$$C = F \cdot \left(\frac{60 \cdot rpm \cdot L_{10}}{10^6} \right)^{\frac{1}{a}} \quad (10.4)$$

10.3.3 CAPACIDADE DE CARGA ESTÁTICA REQUERIDA

A capacidade de carga estática requerida (C_0), de acordo com o catálogo (SKF, 2015), é definida como sendo a carga que produz uma deformação permanente total do corpo rolante e da pista do rolamento no ponto de contato mais carregado igual a 0,0001 do diâmetro do corpo rotante. Ela é calculada por:

$$C_0 = S_0 \cdot F_0 \quad (10.5)$$

Onde F_0 é encontrado da mesma forma representada na Eq. 10.3, porém $X = 0,6$ e $Y = 0,5$.

Quanto ao fator S_0 , (SKF, 2015) recomenda usar os valores:

- $S_0 = 0,5$ quando os serviços forem suaves e sem vibração;
- $S_0 = 1,0$ para serviços com vibração normal;
- $S_0 = 1,5$ a $2,0$ em serviços com intensas cargas de choque.

10.4 CÓDIGO

Os dados de entrada deste código são a vida do rolamento (L), a confiabilidade (R), rotações por minuto (rpm), força radial (F_r) e força axial (F_a). Além disso deve-se informar o fator S_0 conforme as classificações da seção acima e se o mancal é de esfera ou não para determinar o valor de a .

```

#S0 = 0,5 para serviços suaves sem vibração;
#S0 = 1,0 para serviços de vibração normal;
#S0 = 1,5 a 2,0 para serviços com intensas cargas de choque.

S0 = 'serviços de vibração normal'

#responder com s ou n
mancais_esferas = 's'

#carga dinâmica equivalente
F = 1*Fr + 0*Fa

if S0 == 'serviços suaves sem vibração':
    S0 = 0.5
elif S0 == 'serviços de vibração normal':
    S0 = 1.0
elif S0 == 'serviços com intensas cargas de choque':
    S0 = 1.5

if mancais_esferas == 's':
    a = 3
else:
    a = 1/3

#vida nominal
L10 = L / (6.84 * (log(1/R))**0.854 )

#capacidade de carga dinâmica
C = F * ((60*rpm*L10)/10**6)**(1/a)

#capacidade de carga estática
C0 = S0 * ((0.6 * Fr)+(0.5 * Fa))

```

Os dados de saída do código são a vida nominal (L_{10}), a capacidade de carga dinâmica (C) e a capacidade de carga estática (C_{10}).

CONCLUSÃO

Neste trabalho, primeiro foi possível aplicar a linguagem de programação Python nos exercícios da apostila de Elementos de Máquinas I utilizando uma das suas funcionalidades, a automação de processos. Independentemente do nível de complexidade das questões, conseguiu-se deixá-las de uma forma generalista e que fosse possível chegar a uma resposta. Para isso, ao utilizar o código, deve-se inserir os dados de entrada de forma correta.

Porém, em alguns exercícios, mesmo que automatizados, tiveram os valores das variáveis de saída diferentes quando comparados com os encontrados utilizando outra linguagem de programação, por exemplo, o Pascal. Para diminuir essa diferença foi utilizada a precisão dupla com a função Decimal encontrada em alguns códigos.

Ainda assim, foi possível cumprir com sucesso um dos objetivos do trabalho, a automatização. Os exercícios que não passaram por esse processo, foi devido a sua especificidade, a resolução ser rápida ou por não ser complexo. Então, em comum acordo com o orientador, decidiu-se por não fazer neste projeto.

Portanto, eles podem ficar como sugestão para um futuro projeto caso haja interesse de algum aluno de graduação. Junto a isso, pode ser criado um site que hospede a automatização de todos os exercícios.

BIBLIOGRAFIA

- ALBUQUERQUE, O. A. (1980). *ELEMENTOS DE MÁQUINAS*. Guanabara Dois.
- ANDRADE, A. (26 de Novembro de 2022). *Elementos Orgânicos de Máquina 2*. Fonte: Site da UFPR:
<http://www.madeira.ufpr.br/disciplinasalan/AT102-Aula02.pdf>
- BUDYNAS, R. (2011). *ELEMENTOS DE MÁQUINAS DE SHIGLEY PROJETO DE ENGENHARIA MECÂNICA* (8 ed.). PORTO ALEGRE: AMGH. Acesso em 09 de 2022
- Chaveta mecânica: veja para que serve, principais tipos e aplicações*. (22 de Novembro de 2022). Fonte: SITE DA ABEBOM: <https://www.abecom.com.br/chaveta-mecanica/>
- Eixos e buchas estriadas*. (2022 de Novembro de 2022). Fonte: ROLAMENTOS BRASIL:
<https://www.rolamentobrasil.com.br/produtos/eixos/eixos-e-buchas-estriadas>
- Eixos e buchas estriadas*. (25 de Novembro de 2022). Fonte: ROLAMENTOS BRASIL:
<https://www.rolamentobrasil.com.br/produtos/eixos/eixos-e-buchas-estriadas>
- FAIRES, V. M. (1976). *ELEMENTOS ORGÂNICOS DE MÁQUINAS*. Livros Técnicos e Científicos Editora S.A.
- GARCIA, A. (2012). *ENSAIOS DOS MATERIAIS*. Rio de Janeiro: LTC - Livros Técnicos Científicos Editora Ltda.
- HALL, A. (1981). *ELEMENTOS ORGÂNICOS DE MÁQUINAS*. SÃO PAULO: MCGRAW-HILL.
- HIBBELER, R. C. (2010). *RESISTÊNCIA DOS MATERIAIS*. São Paulo: Pearson Education. Acesso em 12 de 08 de 2022
- HIGDON, A. (1981). *MECÂNICA DOS MATERIAIS*. RIO DE JANEIRO: GUANABARA.
- LA PLATA, C. (2022). *APOSTILA DE ELEMENTOS DE MÁQUINAS I*.
- REAL TRATOR PEÇAS*. (26 de Novembro de 2022). Fonte: REAL TRATOR :
<https://www.realtrator.com.br/pecas-para-tratores/arvore-principal-8-velocidade-50x-55x-65x-235-250-265-275-antigos-18-dentes-x-340mm--p>
- SELECTION PARTS*. (23 de Novembro de 2022). Fonte: SELECTION PARTS:
https://www.selectionparts.com.br/MLB-1475298105-eixo-arvore-primaria-cambio-palio-stilo-original-55204974-_JM
- SHIGLEY, J. E. (1984). *ELEMENTOS DE MÁQUINAS I* (1 ed.). RIO DE JANEIRO: LTC. Acesso em SETEMBRO de 2022
- SKF, G. (01 de OUTUBRO de 2015). *CATÁLOGO GERAL ROLAMENTOS*. Fonte: ABECOM:
<https://catalogos.abecom.com.br/catalogos-skf/1-Catalogo-Geral-Rolamentos.pdf>