



Universidade Federal
do Rio de Janeiro
Escola Politécnica

SIMULAÇÃO NUMÉRICA DA TRANSFERÊNCIA DE CALOR EM UM DISCO
DE FREIO DURANTE FRENAGEM USANDO O MÉTODO DE ELEMENTOS
FINITOS

Gabriel da Silva Oliveira Nunes de Aguiar

Projeto de Graduação apresentado ao Curso de Engenharia Mecânica da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro.

Orientador: Gustavo Rabello dos Anjos

Rio de Janeiro
Setembro de 2020



UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

Departamento de Engenharia Mecânica
DEM/POLI/UFRJ



SIMULAÇÃO NUMÉRICA DA TRANSFERÊNCIA DE CALOR EM UM DISCO
DE FREIO DURANTE FRENAGEM USANDO O MÉTODO DE ELEMENTOS
FINITOS

Gabriel da Silva Oliveira Nunes de Aguiar

PROJETO FINAL SUBMETIDO AO CORPO DOCENTE DO DEPARTAMENTO
DE ENGENHARIA MECÂNICA DA ESCOLA POLITÉCNICA DA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE
DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE
ENGENHEIRO MECÂNICO.

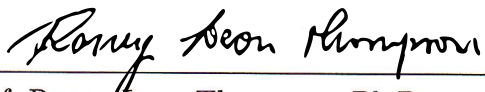
Aprovada por:



Prof. Gustavo Rabello dos Anjos, Ph.D.



Prof. Gustavo César Rachid Bodstein, Ph.D.



Prof. Roney Leon Thompson, Ph.D.

RIO DE JANEIRO, RJ – BRASIL

SETEMBRO DE 2020

Nunes de Aguiar, Gabriel da Silva Oliveira

Simulação Numérica da Transferência de Calor em um Disco de Freio Durante Frenagem usando o Método de Elementos Finitos/ Gabriel da Silva Oliveira Nunes de Aguiar. – Rio de Janeiro: UFRJ/Escola Politécnica, 2020.

XIV, 108 p.: il.; 29, 7cm.

Orientador: Gustavo Rabello dos Anjos

Projeto de Graduação – UFRJ/ Escola Politécnica/ Curso de Engenharia Mecânica, 2020.

Referências Bibliográficas: p. 61 – 63.

1. Frenagem. 2. Elementos Finitos. 3. Simulação.
I. Rabello dos Anjos, Gustavo. II. Universidade Federal do Rio de Janeiro, UFRJ, Curso de Engenharia Mecânica.
III. Simulação Numérica da Transferência de Calor em um Disco de Freio Durante Frenagem usando o Método de Elementos Finitos.

Dedico este trabalho a todos que estiveram comigo ao longo desta caminhada, em especial à minha família, amigos e à Equipe Minerva Baja que me deram todo suporte para concluir essa etapa da minha vida.

Avante Minerva!

Agradecimentos

Sou grato à minha família por sempre me incentivarem e acreditarem que eu seria capaz de superar os obstáculos que a vida me apresentou. Agradeço aos meus amigos que sempre estiveram presentes em qualquer situação e estenderam as mãos quando precisei.

Um agradecimento especial à Equipe Minerva Baja e à todos os integrantes em que tive um enorme prazer de conviver, aprender e ensinar. Sou eternamente grato por todas as experiências e aprendizados que a equipe me proporcionou. As equipes de competição deveriam ser mais valorizadas, pois são extremamente importante para a formação de um aluno de engenharia, tornado-os mais qualificados e abrindo portas para o mercado de trabalho.

Também quero agradecer à Universidade Federal do Rio de Janeiro e a todos os professores do meu curso pela elevada qualidade do ensino oferecido. Em especial ao meu orientador, Gustavo Rabello dos Anjos, pela confiança depositada na minha proposta de projeto e por todo suporte dado nesse trabalho.

Resumo do Projeto de Graduação apresentado à Escola Politécnica/UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro Mecânico

SIMULAÇÃO NUMÉRICA DA TRANSFERÊNCIA DE CALOR EM UM DISCO DE FREIO DURANTE FRENAGEM USANDO O MÉTODO DE ELEMENTOS FINITOS

Gabriel da Silva Oliveira Nunes de Aguiar

Setembro/2020

Orientador: Gustavo Rabello dos Anjos

Programa: Engenharia Mecânica

O presente projeto de graduação consiste no desenvolvimento de uma estrutura computacional em linguagem *Python* para dimensionamento térmico preliminar de um disco de freio baseado na interação disco-pastilha durante uma frenagem de emergência e múltiplas frenagens.

Para atingir este objetivo utilizou-se o Método dos Elementos Finitos (MEF) para resolução da equação axissimétrica de condução do calor em coordenadas cilíndricas associado às condições de contorno desenvolvidas através do balanço de energia no disco de freio. Para derivação das condições de contorno e das equações de transferência de calor, parâmetros como tempo de frenagem, velocidade do veículo, dimensões, geometria e material do disco e pastilha de freio, além da distribuição de pressão na área de contato entre os componentes foram considerados.

O trabalho foi segmentado em duas análises: frenagem de emergência e múltiplas frenagens. Ambas são de extrema importância no dimensionamento térmico do disco de freio de forma a garantir a segurança e desempenho do sistema. Com o objetivo de validar o código numérico realizou-se técnicas como convergência de malha e resultados comparativos baseado em um modelo físico semelhante que possui

solução analítica. Os resultados obtidos foram satisfatórios, o modelo numérico se mostrou coerente com o modelo físico proposto de forma que o objetivo do trabalho foi atingido, obtendo um modelo numérico confiável, estável e acurado para um dimensionamento térmico preliminar do disco de freio.

Abstract of Undergraduate Project presented to POLI/UFRJ as a partial fulfillment of the requirements for the degree of Mechanical Engineer

NUMERICAL SIMULATION OF HEAT TRANSFER IN A BRAKE DISC
DURING BRAKING USING THE FINITE ELEMENT METHOD

Gabriel da Silva Oliveira Nunes de Aguiar

September/2020

Advisor: Gustavo Rabello dos Anjos

Department: Mechanical Engineering

The present graduation project consists of the development of a computational structure in *Python* language for preliminary thermal design of a brake disc based on the disk-pad interaction during an emergency braking and multiple braking.

To achieve this objective, the Finite Element Method (FEM) was used to solve the three-dimensional heat conduction equation in cylindrical coordinates associated with the boundary conditions developed through the energy balance in the brake disc. For derivation of boundary conditions and heat transfer equations, parameters such as braking time, vehicle speed, dimensions, geometry and material of the disc and pad, in addition to the pressure distribution in the contact area between the components were considered .

The work was divided into two analyzes: emergency braking and multiple braking. Both are extremely important in the thermal design of the brake disc in order to guarantee the safety and performance of the system. In order to validate the numerical code, techniques such as mesh convergence and comparative results are performed based on a similar physical model that has an analytical solution. The results obtained were satisfactory, the numerical model was shown to be consistent with the proposed physical model so that the objective of the work was achieved,

obtaining a reliable, stable and accurate numerical model for a preliminary thermal design of the brake disc.

Sumário

Lista de Figuras	xii
Lista de Tabelas	xiv
1 Introdução	1
2 Revisão Bibliográfica	4
2.1 Transferência de Calor	4
2.2 O Método de Elementos Finitos	6
2.3 Sistema de Freios	9
3 Metodologia	12
3.1 Modelo Teórico	12
3.1.1 Modelagem Física	12
3.1.2 Modelagem Matemática	16
3.2 Modelo Numérico	19
3.2.1 Formulação Fraca	19
3.2.2 Discretização Espacial	20
3.2.3 Discretização Temporal	21
3.2.4 Elementos da Malha	22
3.2.5 Forma Matricial	23
4 Algoritmo Computacional	25
4.1 Geração de Malha	25
4.2 Montagem das Matrizes Globais	27
4.3 Aplicação das Condições de Contorno	30
4.4 Algoritmo de Solução	32

4.4.1	Frenagem de Emergência	32
4.4.2	Múltiplas Frenagens	33
5	Validação	35
5.1	Condução de Calor Transiente em um Cilindro Curto	35
5.1.1	Modelo Teórico	35
5.1.2	Modelo Numérico	39
5.1.3	Resultados	40
6	Resultados	48
6.1	Frenagem de Emergência	48
6.2	Múltiplas Frenagens	56
7	Conclusão	59
	Referências Bibliográficas	61
A	Código Fonte - Cilindro Curto	64
A.1	Solução Analítica	64
A.2	Solução Numérica	66
B	Código Fonte - Dimensionamento do Disco de Freio	73
B.1	Frenagem de Emergência	73
B.2	Múltiplas Frenagens	91
C	Dimensionamento do Sistema de Freios	105

Lista de Figuras

1.1	Mecanismo de freio à disco [1]	2
2.1	Exemplo de elementos triangulares (a) e quadriláteros (b) [9]	7
2.2	Exemplo de malhas estruturada (a) e não estruturada (b) [10]	7
2.3	Exemplos de funções de forma em bloco [10]	8
2.4	Montagem das matrizes globais, onde a matriz elementar é associada a cada elemento da malha computacional [10]	9
2.5	Mecanismo de freio à tambor com acionamento hidráulico [1]	10
2.6	Mecanismo de freio à disco com acionamento hidráulico [1]	10
3.1	Esquemático do mecanismo de freio à disco utilizado nos modelos [6]	13
3.2	Condições de contorno no disco durante a frenagem	17
3.3	Condições de contorno no disco durante o tempo de resfriamento	18
3.4	Elemento triangular assiximétrico [10]	22
4.1	Malha bidimensional triangular gerada no <i>Python</i>	27
5.1	Condições de contorno no cilindro curto	37
5.2	Análise de convergência de malha considerando $Bi = 1.0$ e $\tau = 0.4$	42
5.3	Gráficos comparativos entre as soluções analíticas e numéricas para (a) $\tau = 0.4$, (b) $\tau = 0.8$ e (c) $\tau = 1.2$ considerando $Bi = 1.0$	43
5.4	Erro relativo entre as soluções analítica e numérica para $Bi = 1.0$	44
5.5	Gráficos comparativos entre as soluções analítica e numérica para (a) $Bi = 0.01$, (b) $Bi = 0.1$ e (c) $Bi = 1.0$ considerando $\tau = 0.4$	46
5.6	Erro relativo entre as soluções analítica e numérica para $\tau=0.4$	46
5.7	Simulação numérica do cilindro nos instantes (a) $t = 0.1s$, (b) $t = 0.2s$, (c) $t = 0.3s$ e (d) $t = 0.4s$ considerando $Bi = 1.0$ e $\tau = 0.4$	47

6.1	Veículo Baja SAE da Equipe Minerva Baja	48
6.2	Geometria simplificada do mecanismo disco-pastilha de freio e seus principais parâmetros	49
6.3	Análise de convergência de malha para simulação de frenagem de emergência	51
6.4	Visualização da simulação numérica para o disco de freio no plano $z = 2mm$, utilizando o <i>software Paraview</i> , nos instantes (a) $t = 0s$, (b) $t = 0.75s$, (c) $t = 1.5s$ e (d) $t = 2.65s$ considerando $t_b = 2.65s$	52
6.5	Temperatura no disco em diferentes coordenadas radiais em $z = \delta_d$	53
6.6	Temperatura no disco em diferentes coordenadas axiais em função da coordenada radial no tempo $t = t_b$	53
6.7	Temperatura no disco em diferentes coordenadas axiais em $r = 83mm$	54
6.8	Temperatura no disco em diferentes coordenadas radiais ao longo de sua espessura no tempo $t = t_b$	54
6.9	Avanço da temperatura na coordenada axial para frenagem de emergência nos instantes (a) $t = 0s$, (b) $t = 0.75s$, (c) $t = 1.5s$ e (d) $t = 2.65s$ considerando $t_b = 2.65s$	55
6.10	Temperatura máxima no disco de freio em função do número de frenagens n_p para diferentes valores de Δt	57
6.11	Temperatura no disco de freio em função do tempo de simulação para $\Delta t = 20s$	57

Lista de Tabelas

5.1	Parâmetros e propriedades constantes utilizados nas simulações	41
5.2	Parâmetros numéricos utilizados nas simulações	42
5.3	Erros relativos mínimos e máximos considerando $Bi = 1.0$	44
5.4	Constantes A_1 e λ_1 da solução analítica para os valores de Bi especi- ficados	45
5.5	Erros relativos mínimos e máximos considerando $\tau = 0.4$	47
6.1	Parâmetros geométricos utilizados na simulações numéricas	49
6.2	Propriedades dos materiais utilizados na simulações numéricas	49
6.3	Parâmetros do fluido (ar) utilizados na simulações numéricas	50
6.4	Parâmetros de projeto utilizados na simulações numéricas	50
6.5	Parâmetros numéricos utilizados na simulação de frenagem de emergência	50
6.6	Parâmetros adicionais de projeto para a etapa de resfriamento do disco	56

Capítulo 1

Introdução

O desempenho e a segurança dos veículos no decorrer dos anos vem sendo constantemente aprofundado desde o seu surgimento. Novos mecanismos, métodos, materiais, entre outros, são desenvolvidos e analisados com a finalidade de atingir estes objetivos. Um dos sistemas mais importantes do veículo é o sistema de freios que, de acordo com *Limpert* [1], realiza regularmente o controle da velocidade do automóvel em condições de movimento, além do comportamento em trechos retos e curvos com ou sem declividade, garantindo a segurança e integridade do condutor.

O sistema de freios possui três funções básicas [1, 2, 3]:

- Desaceleração do automóvel, incluindo desaceleração total;
- Manutenção da velocidade em situações de declividade;
- Manutenção do veículo estacionário em rampa.

No presente trabalho será analisado somente o primeiro item em trechos retilíneos sob condições de única e múltiplas frenagens. Esse processo envolve a transformação de energia cinética, majoritariamente, em energia térmica durante a frenagem, resultando na elevação da temperatura dos componentes envolvidos. Estaremos interessados em investigar a distribuição de temperatura no disco de freio, uma vez que elevadas temperaturas é o fator principal responsável por falhas no sistema como desgastes prematuros, redução do coeficiente de atrito disco-pastilha, evaporação do fluido de freio e trincas térmicas [4].

A motivação inicial desse trabalho é o desenvolvimento de um modelo numérico capaz de simular o comportamento térmico de um sistema hidráulico de freios à

disco. Essa motivação surgiu da necessidade da Equipe Minerva Baja da Universidade Federal do Rio de Janeiro (UFRJ) em dimensionar e otimizar seus componentes que são constantemente exigidos em competição, de modo a aumentar a confiabilidade do sistema e garantir o desempenho adequado do protótipo.

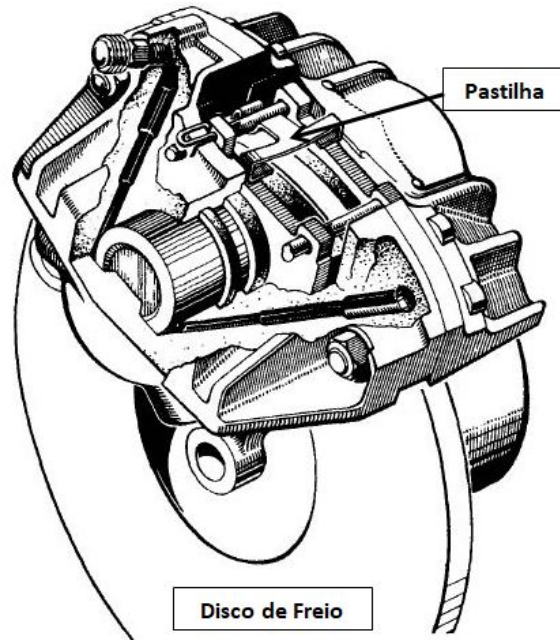


Figura 1.1: Mecanismo de freio à disco [1]

Para desenvolvimento do modelo numérico proposto, utilizou-se a equação axisimétrica da condução do calor em coordenadas cilíndricas [5]. Adotou-se a hipótese que o gradiente na direção tangencial ϕ é desprezível durante a frenagem [6], dessa forma o termo do gradiente na direção tangencial ϕ é retirada da equação. As equações de geração de calor foram obtidas através de um balanço de energia entre o disco e a pastilha de freio, sendo desenvolvido um modelo microscópico para obter a quantidade de energia absorvida pelos dois componentes.

As condições de contorno do problema foram formuladas a partir da interação entre o disco, pastilha e fluido externo, considerando apenas a transferência de calor por condução e convecção, desprezando efeitos da radiação. O disco de freio foi modelado como um corpo sólido axissimétrico com simetria axial, dessa forma, somente metade do disco foi considerado na análise, admitindo-o isolado no plano de simetria.

Para solução do problema foi implementado o Método dos Elementos Finitos uti-

lizando o método de discretização espacial de *Garlekin* no domínio das equações de governo e a discretização temporal pelo Método de Diferenças Finitas. A malha desenvolvida na própria estrutura computacional consistiu em uma malha estruturada orientada no sentido anti-horário com elementos triangulares lineares axissimétricos, modelo adotado e proposto por *Taler et al.* [7]. A solução dos sistemas lineares resultantes do método foi feita pelo método de decomposição LU com o uso da biblioteca *Numpy* e a visualização dos resultados foi realizada através da biblioteca *Matplotlib* e pelo software aberto *Paraview*.

Devido a ausência de modelos analíticos que representem adequadamente o comportamento térmico do problema em questão, no Capítulo 4 é apresentado o desenvolvimento do modelo de um cilindro curto sujeito à efeitos convectivos como modelo de validação da estrutura computacional do presente trabalho.

Esse trabalho foi organizado no seguinte formato:

- Capítulo 1: Introdução
- Capítulo 2: Revisão Bibliográfica
- Capítulo 3: Metodologia
- Capítulo 4: Algoritmo Computacional
- Capítulo 5: Validação
- Capítulo 6: Resultados
- Capítulo 7: Conclusão

Capítulo 2

Revisão Bibliográfica

2.1 Transferência de Calor

Quando nos referimos a energia estamos sujeitos à interpretá-la em duas áreas da ciência: a termodinâmica e a transferência de calor. Segundo *Ösizik* [5] a ciência da termodinâmica trata da relação entre o calor e outras formas de energia, enquanto a ciência da transferência de calor se refere na análise da taxa transferência de calor em um sistema. Em outras palavras, a termodinâmica lida com os estados extremos (inicial e final) do processo ao longo do qual a interação ocorre, não fornecendo informações a respeito da natureza interação ou a taxa que na qual ela ocorre [8].

A ciência da transferência de calor surge como um meio para estudar a natureza dessas interações e quantificar a taxa de transferência de energia no sistema e entre sistemas. Segundo *Incropera et al.* [8], o calor é a energia térmica em trânsito devido uma diferença de temperaturas no espaço, ou seja, sempre que existir uma diferença de temperaturas em um meio ou entre meios haverá, necessariamente, transferência de calor.

Usualmente considera-se a transferência de energia em três modos distintos: condução, convecção e radiação.

A *condução* é o modo de transferência de calor resultante de um gradiente de temperatura em um sistema pelo movimento cinético ou impacto direto de suas moléculas e pelo movimento de elétrons. A energia nesse modo de transferência sempre ocorrerá da região de alta energia para a região de baixa energia. É possível quantificar essa taxa de transferência de energia por condução pela equação empírica

que descreve o fluxo de calor em um sistema denominada **Lei de Fourier**:

$$q_{\text{cond}} = -k\nabla T \quad (2.1)$$

onde k denomina-se condutividade térmica. A condutividade térmica é uma grandeza positiva que depende da natureza do material, podendo variar ou não com a temperatura.

A *convecção* é o modo de transferência de calor resultante da interação entre um sólido e um fluido em movimento. Se a temperatura na superfície do sólido e do fluido forem diferentes, haverá transferência de energia em consequência da movimentação do fluido em relação a superfície. Esse modo de transferência pode ser classificado de acordo com a natureza do escoamento fluido [8]. Nos referimos à convecção forçada quando o escoamento do fluido é originado por meios externos que forcem o fluxo do fluido sobre a superfície do sólido, como ventiladores e bombas. Por outro lado, quando a movimentação do fluido ocorre devido a diferença de densidade causada pela diferença de temperaturas entre o sólido e o fluido, nos referimos à convecção livre (ou natural).

Ainda relacionado a esse modo de transferência de energia, há processos de convecção em que podem ocorrer trocas de calor latente resultando na mudança de fase do fluido. Por mais complexa que seja o fenômeno de transferência de calor por convecção, o fluxo de calor pode ser descrito pela **Lei de Resfriamento de Newton**:

$$q_{\text{conv}} = h(T_s - T_\infty) \quad (2.2)$$

A Eq. 2.2 relaciona a temperatura na superfície T_s e a temperatura do fluido T_∞ com o fluxo de calor por convecção q_{conv} por meio da grandeza h denominada coeficiente de transferência de calor por convecção. Essa grandeza, geralmente, é obtida experimentalmente, pois varia com o tipo de fluxo, geometria do corpo, área de escoamento, entre outros, podendo ser determinado analiticamente nos corpos com geometrias simples [5].

A *radiação térmica* é o modo de transferência de calor natural de um corpo, sólido ou fluido, que possui temperatura diferente de $0K$. A energia é emitida ou absorvida pelo corpo em forma de ondas eletromagnéticas ou na forma de fótons discretos, portanto não necessita de um meio para a sua propagação, diferindo-se dos dois modos anteriores de transferência de energia.

O fluxo de radiação q_{rad} de um corpo real é descrito pela **Lei de Stefan-Boltzmann** associado à emissividade ϵ :

$$q_{\text{rad}} = \epsilon E_b = \epsilon \sigma_b T^4 \quad (2.3)$$

onde E_b é a emitância de um corpo negro, σ é a constante de Boltzmann ($\sigma_b = 5.6697 \times 10^{-8} \frac{W}{m^2.K^4}$) e a emissividade ϵ fica entre zero e a unidade.

2.2 O Método de Elementos Finitos

O Método de Elementos Finitos tem sido largamente aplicado em áreas estruturais e não estruturais da engenharia e da física matemática para encontrar soluções de problemas diferenciais da mecânica do contínuo. Segundo *Logan* [9] o desenvolvimento moderno do método dos elementos finitos começou na década de 1940 no campo da engenharia estrutural com o trabalho de Hrennikoff em 1941 e McHenry em 1943, que usou uma rede de elementos de linha (barras e vigas) para a solução de tensões em sólidos contínuos. Em um artigo publicado em 1943, mas não amplamente reconhecido por muitos anos, Courant propôs configurar a solução de tensões em uma forma variacional em um domínio discretizado por elementos triangulares.

Posteriormente, Galerkin utiliza o *Método do Resíduo Ponderado* na determinação das constantes da formulação variacional onde as mesmas funções bases foram utilizadas nas funções peso. Este procedimento fica conhecido como *Formulação de Galerkin* e é largamente utilizado até hoje.

Após a formulação de Galerkin, o método de elementos finitos começou a ser aplicado nos problemas envolvendo dinâmica de fluidos e transferência de calor, principalmente em problemas envolvendo geometrias, carregamentos e propriedades complexas na qual uma solução analítica proveniente da resolução de equações diferenciais são difíceis de serem obtidas.

O Método de Elementos Finitos é um método numérico cujo conceito é participar o domínio do meio contínuo em inúmeros sub-domínios finitos discretos denominados elementos conectados por pontos denominados pontos nodais (ou nós), e obter uma solução aproximada aplicando a equação em cada elemento e resolvendo sistemas lineares simples [10]. Esse domínio discreto é conhecido como malha, na qual a sua geração pode ser realizada em qualquer geometria. Há uma variedade de

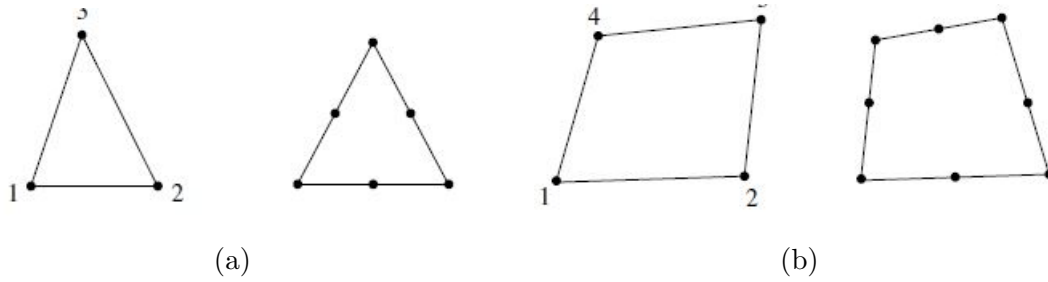


Figura 2.1: Exemplo de elementos triangulares (a) e quadriláteros (b) [9]

elementos que podem ser utilizadas na discretização do domínio, como ilustrado na Figura 2.1, não necessitando de malha estruturada (número de vizinhos por nó é constante) e uniforme (espaçamentos constantes), o que caracteriza uma vantagem do método em relação à outros.

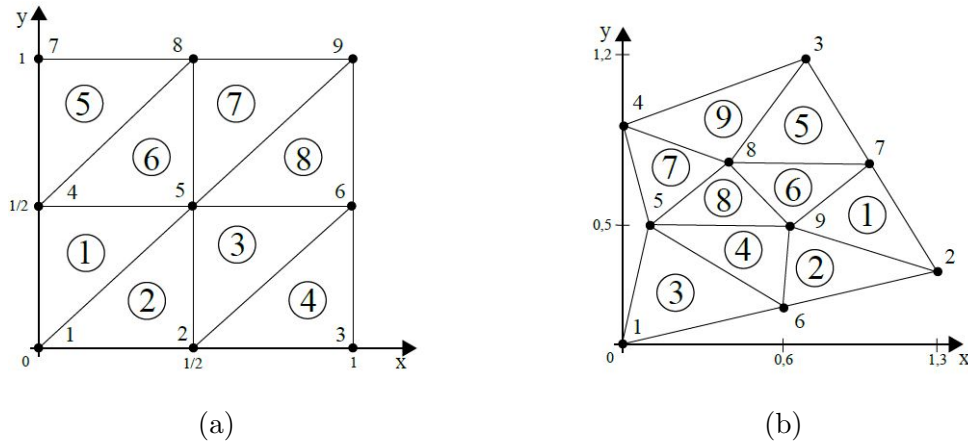


Figura 2.2: Exemplo de malhas estruturada (a) e não estruturada (b) [10]

A processo de aplicação do método de elementos finitos, em síntese, se inicia com a transformação da equação diferencial original (forma forte) para a sua forma fraca ou variacional em seu domínio. Essa primeira etapa consiste na utilização do **Método do Resíduo Ponderado**, no qual é necessário encontrar uma solução aproximada que satisfaça a seguinte relação:

$$\int_{\Omega} w(x) f''(x) d\Omega = 0 \quad (2.4)$$

onde $w(x)$ é denominado função peso, $f''(x)$ é a equação diferencial original genérica de segunda ordem e Ω é o domínio do problema. Nessa etapa utiliza-se a integração por partes pelo **Teorema de Green** para redução dos termos de ordens superiores.

A segunda etapa consiste na discretização do domínio Ω e da escolha da aproximação para as funções teste e peso. A aproximação mais utilizada e adotada nesse trabalho foi a aproximação de *Galerkin*, que busca aproximar as funções $w(x)$ e $f(x)$ como:

$$w^e(x) = \sum_{i=1}^{np} w_i N_i^e(x) \quad (2.5)$$

$$f^e(x) = \sum_{j=1}^{np} f_j N_j^e(x) \quad (2.6)$$

onde w_i e f_j são constantes a serem determinadas e no Método de Galerkin, as funções as funções de forma $N_i(x)$ e $N_j(x)$ são iguais. A aproximação ocorre em cada elemento da malha, composta por n_p pontos nodais e n_e elementos.

A Figura 2.3 ilustra dois tipos de funções de forma para discretização do problema, diferindo-se entre si por meio do grau do polinômio interpolador. Segundo *Anjos* [10] uma função de forma de ordem mais alta não necessariamente irá fornecer resultados mais precisos, pois pode trazer instabilidades numéricas e aumento significativo dos recursos computacionais.

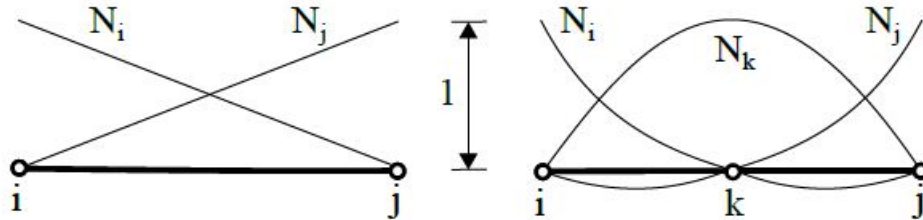


Figura 2.3: Exemplos de funções de forma em bloco [10]

Em seguida, as funções de aproximação são aplicadas na formulação fraca da equação que, por sua vez, são aplicadas a cada elemento da malha para obter as matrizes do sistema local. Posteriormente, realiza-se a superposição das matrizes do sistema local para obter as matrizes do sistema global representativa do domínio Ω , como mostrado na Figura 2.4.

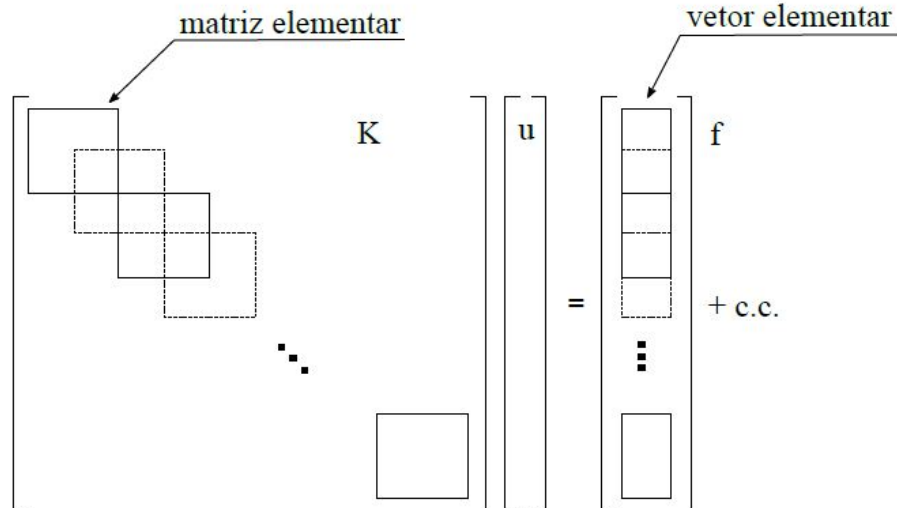


Figura 2.4: Montagem das matrizes globais, onde a matriz elementar é associada a cada elemento da malha computacional [10]

Com o sistema montado, aplica-se as condições de contorno e resolve-se o sistema linear através de algum método de solução de sistemas lineares. O procedimento aplicado nesse trabalho está detalhado no Capítulo 3.

2.3 Sistema de Freios

Com o advento e aprimoramento das rodas, os veículos passaram a transportar progressivamente mais carga de forma mais rápida e com menor gasto de energia [11]. Esse feito despertou a atenção dos projetista a se preocuparem cada vez mais com os procedimentos de frenagem de forma a desenvolver mecanismos de frenagens capazes de parar o automóvel no tempo ou distância desejável de forma eficiente, segura e exigindo baixa necessidade de manutenção.

A utilização dos freios é tão antiga quanto os das rodas. De acordo com *Nicolazzi* [11], os primeiros freios industriais possuíam cintas de aço envolvendo tambores, também de aço. As cintas eram fixas por pinos e o acionamento das mesmas eram realizadas de forma mecânica por meio de alavancas. O sistema era ineficiente com elevados desgastes dos componentes. Posteriormente, esse mecanismo sofreu um processo de otimização com mudanças nos materiais das cintas, agora no formato de sapatas, e dos tambores, bem como o tipo de acionamento que se tornou hidráulico.

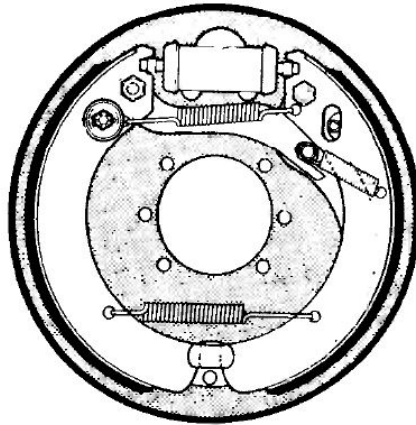


Figura 2.5: Mecanismo de freio à tambor com acionamento hidráulico [1]

Os veículos modernos adotam outro tipo de mecanismo de frenagem: o mecanismo de freio à disco. Esse mecanismo é ilustrado na Figura 2.6 conjuntamente com o sistema de freio à tambor.

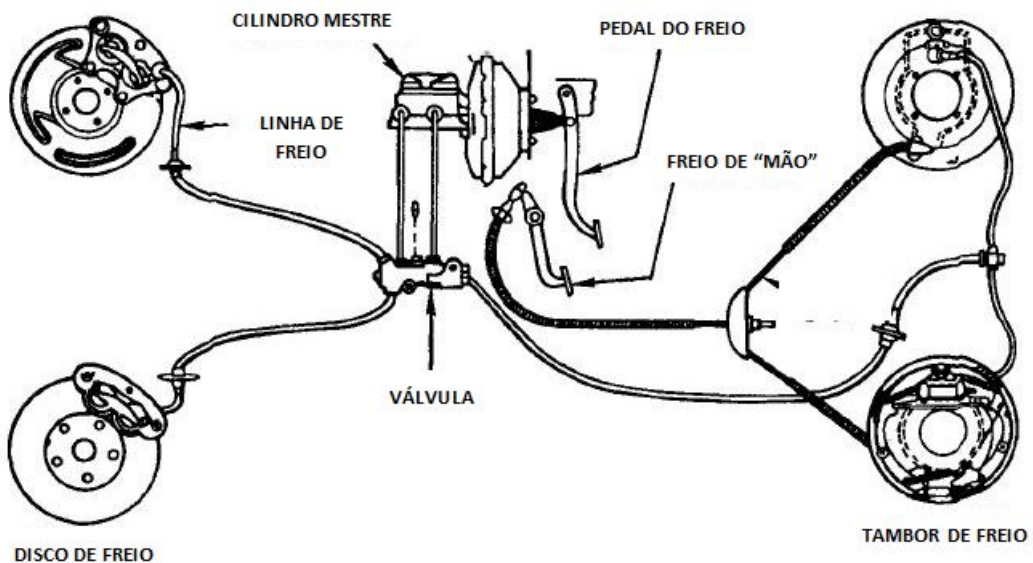


Figura 2.6: Mecanismo de freio à disco com acionamento hidráulico [1]

O funcionamento do mecanismo de freio à disco ocorre da seguinte forma [2]:

- O condutor aciona o pedal de freio (alavanca) aplicando uma determinada carga. Essa carga é ampliada e aplicada no pistão do cilindro mestre fazendo-o deslocar;

- O deslocamento do pistão produz um deslocamento de fluido de freio, produzindo uma pressão inicial;
- Essa pressão percorre as linhas de freio com determinada eficiência já que ocorre perda de pressão nas linhas;
- O fluido pressurizado entra em contato com o êmbolo das pinças, que pelo Princípio de Pascal, amplia a pressão do sistema;
- Por fim, o êmbolo aciona a pastilha produzindo uma força normal no disco e, conseqüentemente, uma força de atrito capaz de parar o movimento do disco.

A energia cinética do veículo é, majoritariamente, convertida em energia térmica (ou calor) que será absorvido tanto pelo disco quanto pelas pastilhas de freio. Nesse trabalho não estamos interessados em desenvolver o equacionamento da dinâmica de frenagem, por outro lado, as variáveis e dimensionamento do projeto de freio que serão necessários para simulação e verificação da estrutural computacional desenvolvida foram fornecidos pela Equipe Minerva Baja e se encontram no **Apêndice C**.

Capítulo 3

Metodologia

Nesse capítulo será iniciado o desenvolvimento dos modelos para frenagem de emergência e múltiplas frenagens. Durante o desenvolvimento, quando necessário, será distinguido os parâmetros entre os dois modelos. Vale ressaltar que o modelo de múltiplas frenagens é considerado uma generalização do modelo para frenagem de emergência [12], portanto grande parte dos parâmetros são compartilhados. Para melhor interpretação o capítulo foi dividido em:

- **Modelo Teórico:** será apresentado as hipóteses e parâmetros dos modelos, desenvolvimento das equações de governo e condições de contorno, além das equações de fluxo de calor para o disco de freio como desenvolvido em [4, 6, 13, 14, 15, 16, 17].
- **Modelo Numérico:** será desenvolvido e aplicado o Método de Elementos Finitos nos modelos, abordando a formulação da forma fraca da equação de governo, o método de discretização espacial de *Galerkin*, o método de discretização temporal, a formulação na forma matricial, além do elemento finito utilizado na malha.

3.1 Modelo Teórico

3.1.1 Modelagem Física

Os freios, em sua essência, são mecanismo de transformação de energia. Quando ocorre a frenagem de um veículo, a energia cinética do veículo é transferida na forma

de energia térmica (ou calor) para o disco e para a pastilha de freio por condução e parte dessa energia é dissipada para o fluido ambiente através da transferência de calor por convecção. A Figura 3.1 ilustra o mecanismo de freio à disco utilizado na formulação do problema. O mecanismo adotado consiste em dois elementos: um disco rotativo axissimétrico e uma pastilha estática não axissimétrica.

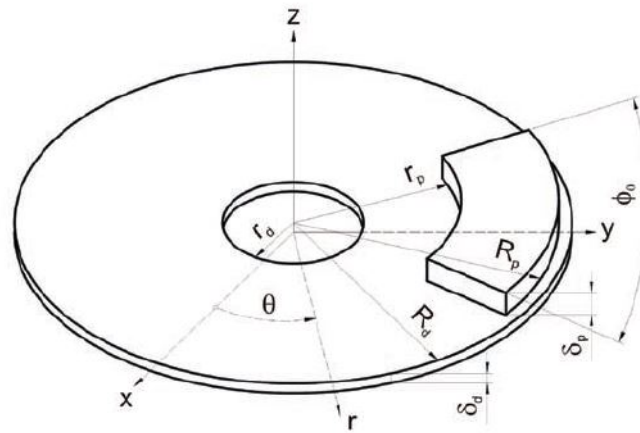


Figura 3.1: Esquemático do mecanismo de freio à disco utilizado nos modelos [6]

Para a formulação dos modelos, algumas hipóteses foram adotadas:

- A área de contato (ou deslizamento) entre disco e pastilha são consideradas iguais. Nessa área a pressão é distribuída de forma uniforme;
- Os materiais são considerados isotrópicos, ou seja, as propriedades dos materiais são constantes e independentes da variação de temperatura ao longo de uma seção;
- O fluxo de calor gerado pelo atrito entre pastilha-disco é dissipado para o ambiente em regiões de contato do fluido (ar) com o mecanismo através da *Lei de Resfriamento de Newton*.
- A temperatura ambiente T_∞ é constante;
- Considerou-se simetria axial do disco de freio, assim somente metade do disco é considerado no modelo e o plano de simetria é considerado isolado;
- O coeficiente de transferência de calor por convecção h varia com o tempo e com a coordenada radial do disco, conforme sugerido em *Limpert* [1];

- A dissipação de calor por radiação é negligenciada devido baixos tempos de frenagem e baixas temperaturas atingidas durante as frenagens [4, 6, 13, 14, 15, 16, 17];
- A desaceleração do veículo durante a frenagem é considerada constante.
- Para o modelo de múltiplas frenagens, considera-se aceleração constante até a velocidade máxima do veículo, mantendo-o constante durante o tempo de resfriamento do sistema estipulado nas simulações.

Além dessa hipótese iniciais é importante ressaltar que os componentes do sistema possuem resistência térmicas diferentes originando em um contato térmico imperfeito. Portanto, parte da energia térmica é absorvida pelo disco e a outra parte, pela pastilha e, essa partição de energia é representada pelo coeficiente de partição de calor σ [1, 4, 6, 13, 14, 15, 17]:

$$\sigma = \frac{\zeta_d S_d}{\zeta_d S_d + \zeta_p S_p} \quad (3.1)$$

onde ζ_d e ζ_p são denominados efusividades térmicas e S_d e S_p são as superfícies de contato do disco e da pastilha, respectivamente.

A efusividade térmica é definido como [4, 6]:

$$\zeta = \sqrt{k\rho c} \quad (3.2)$$

onde ρ e c são a densidade e o calor específico do material, respectivamente.

As superfícies de contato do disco e da pastilha são definidas como [4]:

$$S_d = 2\pi r dr \quad (3.3)$$

$$S_p = \phi_0 r dr \quad (3.4)$$

onde ϕ_0 é o ângulo de contato da pastilha e r é a distância radial em relação ao centro do disco.

O coeficiente de partição de calor ilustrado acima é utilizado para a situação de uma única frenagem ou frenagem de emergência. Segundo *Limpert* [1], devido às altas temperaturas atingidas para determinados tempos de resfriamento do sistema, o coeficiente de partição de calor $\bar{\sigma}$ para a situações de múltiplas frenagens é definido como:

$$\bar{\sigma} = \left[1 + \frac{k_p S_p}{S_d (k_p + h \delta_p)} \right]^{-1} \quad (3.5)$$

onde δ_p é a espessura da pastilha e o coeficiente de transferência de calor por convecção para discos sólidos é definido [1] como:

$$h = 0.70 \left(\frac{2k_a}{R_d} \right) Re^{0.55}, \quad Re \leq 2.4 \times 10^5 \quad (3.6)$$

$$h = 0.04 \left(\frac{2k_a}{R_d} \right) Re^{0.80}, \quad Re \geq 2.4 \times 10^5 \quad (3.7)$$

onde k_a é a condutividade térmica do ar, R_d é o raio externo do disco e Re é o número de Reynolds definido como:

$$Re(r, t) = \frac{4\omega(t)r^2}{\nu} \quad (3.8)$$

onde ν é a viscosidade cinemática do ar e ω é a velocidade angular do disco. Pela hipótese de desaceleração constante, a velocidade angular do disco decresce de ω_0 até o repouso no tempo de frenagem t_b pela seguinte relação:

$$\omega(t) = \omega_0 \left(1 - \frac{t}{t_b} \right), \quad t \leq t_b \quad (3.9)$$

De forma análoga, para múltiplas frenagens, pela hipótese de aceleração constante, o disco atinge sua velocidade angular máxima ω_0 no tempo de aceleração t_a através da seguinte relação:

$$\omega(t) = \omega_0 \left(\frac{t}{t_a} \right), \quad t \leq t_a \quad (3.10)$$

$$\omega = \omega_0, \quad t \geq t_a \quad (3.11)$$

visto que quando o veículo atinge a velocidade máxima, por hipótese, esse se mantém constante até o tempo de resfriamento estipulado pelo projetista.

Para cálculo do fluxo de calor originado pelo atrito devido o contato deslizante da pastilha com o disco de freio é necessário definir a distribuição de pressão e a taxa de quantidade de calor produzida. Segundo *Talati et al* [4], existem dois modelos para distribuição de pressão:

- **Pressão Uniforme:** a pressão aplicada em qualquer região da superfície de contato do disco e pastilha é igual à pressão máxima, ou seja,

$$p = p_{max} \quad (3.12)$$

- **Desgaste Uniforme:** a pressão aplicada é função da coordenada radial do disco e a pressão máxima ocorre no raio interno r_p onde a pastilha é posicionada. Portanto,

$$p = p_{max} \left(\frac{r_p}{r} \right) \quad (3.13)$$

O primeiro modelo de distribuição de pressão ocorre quando as pastilhas estão novas, enquanto o segundo modelo será mais realístico quando ocorre desgastes das pastilhas após várias frenagens [4].

A taxa de calor total produzida durante uma frenagem em um elemento de área de disco e pastilha é definido como:

$$d\dot{E} = d\dot{E}_p + d\dot{E}_d = (1 - \sigma)dP + \sigma dP \quad (3.14)$$

$$dP = \omega r dF_{at} = \mu \omega(t) p \phi_0 r^2 dr \quad (3.15)$$

onde μ é o coeficiente de atrito disco-pastilha e F_{at} é a força de atrito. Para obter o fluxo de calor q nos componentes, divide-se a taxa de calor absorvida por cada componente por sua superfície de contato. Como nesse trabalho estamos interessados somente no dimensionamento do disco, temos:

$$q(r, t) = \frac{d\dot{E}_d}{dS_d} = \frac{\phi_0}{2\pi} \mu \sigma p r \omega(t) \quad (3.16)$$

Para a distribuição de pressão foi escolhido o modelo de pressão uniforme pois fornece valores mais elevados de fluxo de calor em relação ao modelo de desgaste uniforme. Desse modo, fornece maior segurança e confiabilidade para o dimensionamento do sistema. Portanto, o fluxo de calor q será função da coordenada radial e do tempo:

$$q(r, t) = \frac{d\dot{E}_d}{dS_d} = \frac{\phi_0}{2\pi} \mu \sigma p_{max} r \omega(t) \quad (3.17)$$

Vale ressaltar que para múltiplas frenagens o coeficiente de partição utilizado será o $\bar{\sigma}$ definido na Eq. 3.5.

3.1.2 Modelagem Matemática

Para representar o campo de temperatura no disco de freio durante a frenagem é necessário uma formulação matemática que governe o fenômeno. Os efeitos dos fenômenos de condução em um sólido é descrito pela **equação de condução do calor**.

Para descrever o fenômeno de condução de calor em um disco de freio foi considerado a equação formulada em coordenadas cilíndricas (r, z, θ) centrado no eixo do disco e z apontando para espessura do disco $2\delta_d$, como se segue:

$$\frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \frac{\partial^2 T}{\partial r} + \frac{1}{r^2} \frac{\partial^2 T}{\partial \theta^2} + \frac{\partial^2 T}{\partial z^2} = \frac{1}{\alpha_d} \frac{\partial T}{\partial t} \quad (3.18)$$

$$r_d \leq r \leq R_d, \quad 0 \leq \theta \leq 2\pi, \quad 0 \leq z \leq \delta_d, \quad t > 0$$

Segundo *Kuciej et al.* [6], o número de Peclet em disco automotivos geralmente é da ordem de 10^5 , conseqüentemente, o fluxo de calor na direção tangencial θ pode ser considerado uniforme. Portanto, o gradiente possui uma direção preferencial, de modo que toda circunferência concêntrica ao disco é uma superfície equipotencial. Logo, a equação de condução do calor no disco se reduz a:

$$\frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \frac{\partial T}{\partial r} + \frac{\partial^2 T}{\partial z^2} = \frac{1}{\alpha_d} \frac{\partial T}{\partial t}, \quad r_d \leq r \leq R_d, \quad 0 \leq z \leq \delta_d, \quad t > 0 \quad (3.19)$$

$$\alpha_d = \frac{k_d}{\rho_d c_d} \quad (3.20)$$

onde α_d é a difusividade térmica do disco.

A equação de condução do calor para o disco necessita de condições de contorno e de uma condição inicial para sua solução. A Figura 3.2 ilustra um esquemático das condições de contorno do problema térmico para melhor compreensão.

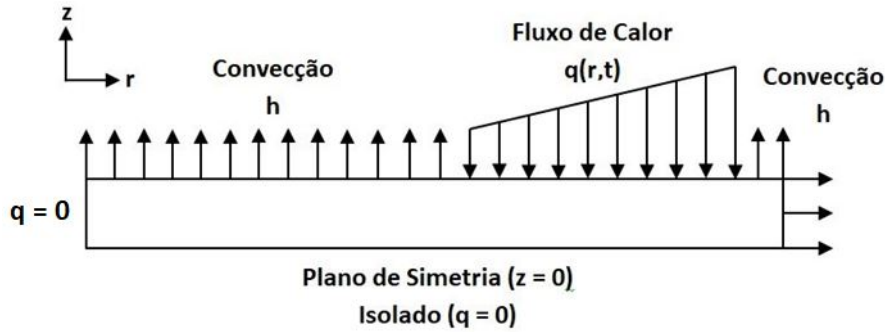


Figura 3.2: Condições de contorno no disco durante a frenagem

A condições de contorno no disco são obtidas através de uma balanço de energia nas superfícies. Portanto, as condições de contorno foram formuladas da seguinte

forma:

$$\left. \frac{\partial T}{\partial z} \right|_{z=0} = 0, \quad r_d \leq r \leq R_d, \quad t > 0 \quad (3.21)$$

$$\left. \frac{\partial T}{\partial r} \right|_{r=r_d} = 0, \quad 0 \leq z \leq \delta_d, \quad t > 0 \quad (3.22)$$

$$\left. \frac{\partial T}{\partial z} \right|_{z=\delta_d} = H_d [T_\infty - T(r, \delta_d, t)], \quad r_d \leq r \leq r_p \wedge R_p \leq r \leq R_d, \quad t > 0 \quad (3.23)$$

$$\left. \frac{\partial T}{\partial z} \right|_{z=\delta_d} = -\frac{1}{k_d} q(r, \delta_d, t), \quad r_p \leq r \leq R_p, \quad t > 0 \quad (3.24)$$

$$\left. \frac{\partial T}{\partial r} \right|_{r=R_d} = H_d [T_\infty - T(R_d, z, t)], \quad 0 \leq z \leq \delta_d, \quad t > 0 \quad (3.25)$$

$$H_d = \frac{h}{k_d} \quad (3.26)$$

Considerou-se que o disco, inicialmente, estava à temperatura ambiente T_0 , portanto a condição inicial é definida como:

$$T(r, z, 0) = T_0, \quad r_d \leq r \leq R_d, \quad 0 \leq z \leq \delta_d \quad (3.27)$$

Na situação de múltiplas frenagens, nos instantes em que o veículo está acelerando e mantém sua velocidade durante o tempo de resfriamento, a condição de contorno da Eq. 3.24 é alterada, pois o fluxo de calor é nulo. Em compensação, considerou-se transferência de calor por convecção. A Figura 3.2 ilustra as novas condições de contorno nessa situação e a nova condição de contorno na região ($r_p \leq r \leq R_p$) é definida a seguir.

$$\left. \frac{\partial T}{\partial z} \right|_{z=\delta_d} = H_d [T_\infty - T(r, \delta_d, t)], \quad r_p \leq r \leq R_p, \quad t > 0 \quad (3.28)$$

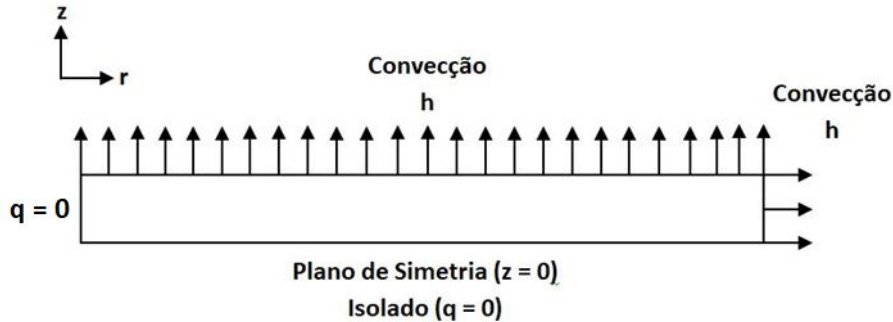


Figura 3.3: Condições de contorno no disco durante o tempo de resfriamento

Modelos analíticos, como mostrados em [4, 6] foram desenvolvidos para esse problema térmico. No entanto, a solução da equação do calor para o disco requer

ferramentas matemáticas complexas, o que dificulta a aplicação em situações reais e geometrias complexas. Por esse motivo, decidimos pela aplicação do Método de Elementos Finitos, desenvolvido na seção a seguir.

3.2 Modelo Numérico

3.2.1 Formulação Fraca

A formulação forte do modelo é o conjunto de suas equações diferenciais aplicadas à um domínio, e de suas condições de contorno aplicadas à fronteira desse domínio. Essa formulação foi apresentada no capítulo anterior nas Eqs. 3.19 à 3.28.

Para o desenvolvimento do método de elementos finitos, como abordado no Capítulo 2, a primeira etapa do processo é a redução da forma forte da equação para a sua forma fraca ou variacional. A formulação fraca da equação é obtida através da ponderação das equações na forma forte integradas no domínio, no qual o fator de ponderação é função peso $\omega(r, z)$. O objetivo é encontrar uma aproximação da solução de modo a produzir o menor resíduo possível [18, 19, 20, 21].

Assim sendo, buscaremos forçar o resíduo equivalente a zero em um sentido médio [18], logo:

$$\int_{\Omega} \omega(r, z) \left[\frac{\partial^2 T(r, z, t)}{\partial r^2} + \frac{1}{r} \frac{\partial^2 T(r, z, t)}{\partial r} + \frac{\partial^2 T(r, z, t)}{\partial z^2} - \frac{1}{\alpha_d} \frac{\partial T(r, z, t)}{\partial t} \right] d\Omega = 0 \quad (3.29)$$

onde Ω é o domínio espacial:

$$T : \Omega \rightarrow \mathbb{R}, \quad r_d \leq r \leq R_d \cup 0 \leq z \leq \delta_d \quad (3.30)$$

Desenvolvendo a integral, obtemos duas parcelas cuja a derivada é de segunda ordem:

$$\int_{\Omega} \omega(r, z) \frac{\partial^2 T(r, z, t)}{\partial r^2} d\Omega \quad (3.31)$$

$$\int_{\Omega} \omega(r, z) \frac{\partial^2 T(r, z, t)}{\partial z^2} d\Omega \quad (3.32)$$

Nessas parcelas aplicaremos o Teorema de Green com o intuito de reduzir a ordem da derivada e separar os termos avaliados no contorno Γ e no domínio Ω .

Assim, temos:

$$\int_{\Omega} \omega \frac{\partial^2 T}{\partial r^2} d\Omega = \int_{\Gamma} \omega \frac{\partial T}{\partial r} d\Gamma - \int_{\Omega} \frac{\partial \omega}{\partial r} \frac{\partial T}{\partial r} d\Omega \quad (3.33)$$

$$\int_{\Omega} \omega \frac{\partial^2 T}{\partial z^2} d\Omega = \int_{\Gamma} \omega \frac{\partial T}{\partial z} d\Gamma - \int_{\Omega} \frac{\partial \omega}{\partial z} \frac{\partial T}{\partial z} d\Omega \quad (3.34)$$

Portanto, a equação de condução de calor no disco de freio em sua forma fraca é representada por:

$$- \int_{\Omega} \left[\frac{\partial \omega}{\partial r} \frac{\partial T}{\partial r} + \frac{\partial \omega}{\partial z} \frac{\partial T}{\partial z} - \frac{\omega}{r} \frac{\partial T}{\partial r} + \frac{\omega}{\alpha_d} \frac{\partial T}{\partial t} \right] d\Omega + \int_{\Gamma} \omega \frac{\partial T}{\partial r} d\Gamma + \int_{\Gamma} \omega \frac{\partial T}{\partial z} d\Gamma = 0 \quad (3.35)$$

Podemos reescrever as parcelas avaliadas no contorno Γ , dividindo o contorno em duas regiões Γ_h e Γ_q da seguinte forma:

$$\int_{\Gamma} \omega \frac{\partial T}{\partial r} d\Gamma + \int_{\Gamma} \omega \frac{\partial T}{\partial z} d\Gamma = H_d \int_{\Gamma_h} \omega (T_{\infty} - T) d\Gamma_h - \frac{1}{k_d} \int_{\Gamma_q} \omega q d\Gamma_q \quad (3.36)$$

onde Γ_h e Γ_q são as superfícies na qual há convecção e fluxo de calor, respectivamente.

Por fim, substituindo a Eq. 3.36 na Eq. 3.35 e reordenando os termos, obtemos a formulação fraca da equação que governa o problema térmico:

$$\begin{aligned} & - \int_{\Omega} \left[\frac{\partial \omega}{\partial r} \frac{\partial T}{\partial r} + \frac{\partial \omega}{\partial z} \frac{\partial T}{\partial z} - \frac{\omega}{r} \frac{\partial T}{\partial r} \right] d\Omega + H_d \int_{\Gamma_h} \omega T d\Gamma_h \\ & + \frac{1}{\alpha_d} \int_{\Omega} \omega \frac{\partial T}{\partial t} d\Omega + H_d \int_{\Gamma_h} \omega T_{\infty} d\Gamma_h - \frac{1}{k_d} \int_{\Gamma_q} \omega q d\Gamma_q = 0 \end{aligned} \quad (3.37)$$

onde ressalta-se que a parcela avaliada no contorno Γ_q para a situação de múltiplas frenagens no decorrer do tempo de aceleração e resfriamento é nula.

3.2.2 Discretização Espacial

Nesse trabalho utilizaremos a formulação de **Galerkin** para discretizar o domínio espacial. Esse método consiste em adotar a mesma função de forma $N(r, z)$ para função peso $\omega(r, z)$ e a função de aproximação da variável à ser determinada $T(r, z, t)$, isto é:

$$\omega^e(r, z) \cong \sum_{i=1}^{n_p} N_i^e(r, z) a_i \quad (3.38)$$

$$T^e(r, z) \cong \sum_{j=1}^{n_p} N_j^e(r, z) b_j \quad (3.39)$$

onde $N_i^e = N_j^e$, n_p é o número de pontos nodais da malha e a_i e b_j são constantes a serem determinadas. Aplica-se essa discretização em cada elemento da malha composta por n_e elementos.

Aplicando as Eqs. 3.38 e 3.39 na formulação fraca descrita na Eq. 3.37, temos:

$$\begin{aligned} \frac{1}{\alpha_d} \sum_i \sum_j \int_{\Omega} N_i^e N_j^e d\Omega \frac{\partial b_j}{\partial t} + \left[\sum_i \sum_j \int_{\Omega} \left(\frac{\partial N_i^e}{\partial r} \frac{\partial N_j^e}{\partial r} + \frac{\partial N_i^e}{\partial z} \frac{\partial N_j^e}{\partial z} - \frac{N_i^e}{r} \frac{\partial N_j^e}{\partial r} \right) d\Omega \right. \\ \left. + \sum_i \sum_j \int_{\Gamma_h} H_d N_i^e N_j^e d\Gamma_h \right] b_j = -\frac{1}{k_d} \sum_i \int_{\Gamma_q} N_i^e q d\Gamma_q + H_d \sum_i \int_{\Gamma_h} N_i^e T_{\infty} d\Gamma_h \end{aligned} \quad (3.40)$$

A Eq. 3.40 pode ser representada na forma matricial:

$$[C_{ij}] \left\{ \frac{\partial b_j}{\partial t} \right\} + [K_{ij}] [b_j] = \{R_i\} \quad (3.41)$$

onde $[C]$ é denominada matriz de massa ou capacidade térmica, $[K]$ é a matriz rigidez ou condutividade térmica e $\{R\}$ é o vetor força térmica.

3.2.3 Discretização Temporal

Para a derivada parcial temporal $\partial b_j / \partial t$ diversos esquemas de discretização podem ser usadas, dentre eles o próprio método de elementos finitos, ou então o método de diferenças finitas [10]. O esquema aplicado nesse trabalho foi uma abordagem geral de discretização por diferenças finitas, isto é:

$$\{b_j\}_{t+\Delta t} = \{b_j\}_t + \left[(1 - \beta) \left\{ \frac{\partial b_j}{\partial t} \right\}_t + \beta \left\{ \frac{\partial b_j}{\partial t} \right\}_{t+\Delta t} \right] \quad (3.42)$$

$$\left\{ \frac{\partial b_j}{\partial t} \right\} = \frac{\{b_j\}_{t+\Delta t} - \{b_j\}_t}{\Delta t} \quad (3.43)$$

onde β é um fator que varia de zero à unidade. Quando β é nulo, o método explícito é aplicado, quando β é igual a 0.5, o método de Crank-Nicolson é aplicado e, por fim, quando β é igual à unidade, o método implícito é aplicado.

Aplicando as Eqs. 3.42 e 3.43 na forma matricial da Eq. 3.41, obtemos a formulação do problema térmico do disco em sua forma matricial completa:

$$([C] + \beta \Delta t [K]) \{b_j\}_{t+\Delta t} = ([C] - (1 - \beta) \Delta t [K]) \{b_j\}_t + \{R\} \quad (3.44)$$

Na solução desse trabalho foi utilizado a discretização proposta por **Crank-Nicolson** pois por ser um méto semi-implícito e de segunda ordem proporciona uma estabilidade maior à solução numérica.

3.2.4 Elementos da Malha

A seleção adequada dos elementos da malha e da ordem do polinômio interpolador são essenciais para uma boa precisão da solução. Para problemas que não envolvem acoplamento entre variáveis em geometrias bidimensionais axissimétricas, como no presente trabalho, elementos triangulares assiximétricos com polinômios de interpolação (ou função de forma) lineares são comumente utilizados porque possibilitam uma boa discretização de superfícies irregulares devido a sua simplicidade geométrica [18, 22].

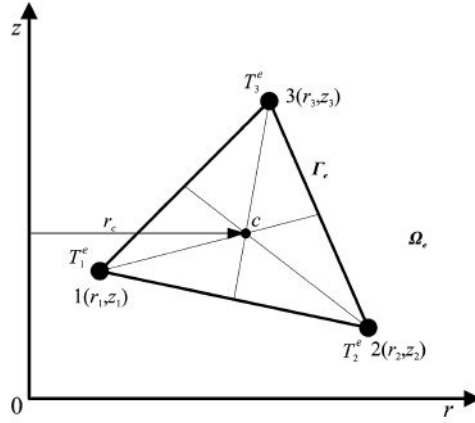


Figura 3.4: Elemento triangular assiximétrico [10]

Para esse tipo de elemento linear, as funções de forma são definidas em cada ponto nodal ($i, j = 3$) que forma o elemento como [7, 19]:

$$N_1^e = \frac{1}{2A^e}(a_1^e + b_1^e z + c_1^e r) \quad (3.45)$$

$$N_2^e = \frac{1}{2A^e}(a_2^e + b_2^e z + c_2^e r) \quad (3.46)$$

$$N_3^e = \frac{1}{2A^e}(a_3^e + b_3^e z + c_3^e r) \quad (3.47)$$

onde A^e é área do elemento e a , b e c são constantes, definidos como:

$$A^e = \frac{1}{2} \begin{vmatrix} 1 & r_1 & z_1 \\ 1 & r_2 & z_2 \\ 1 & r_3 & z_3 \end{vmatrix} \quad (3.48)$$

$$\begin{aligned} a_1^e &= r_2 z_3 - r_3 z_2, & b_1^e &= z_2 - z_3, & c_1^e &= r_3 - r_2 \\ a_2^e &= r_3 z_1 - r_1 z_3, & b_2^e &= z_3 - z_1, & c_2^e &= r_1 - r_3 \\ a_3^e &= r_1 z_2 - r_2 z_1, & b_3^e &= z_1 - z_2, & c_3^e &= r_2 - r_1 \end{aligned} \quad (3.49)$$

3.2.5 Forma Matricial

Segundo *Lewis* [18] e *Jayas* [22], as matrizes e vetores definidas na equação 3.41 podem ser rescritas de forma matricial da seguinte forma:

$$[\mathbf{C}^e] = \frac{1}{\alpha_d} \int_{\Omega} [\mathbf{N}^e]^T [\mathbf{N}^e] d\Omega \quad (3.50)$$

$$[\mathbf{K}^e] = [K_c^e] + [K_h^e] = \int_{\Omega} [\mathbf{B}^e]^T [\mathbf{D}^e] [\mathbf{B}^e] d\Omega + H_d \int_{\Gamma_h} [\mathbf{N}^e]^T [\mathbf{N}^e] d\Gamma_h \quad (3.51)$$

$$\{\mathbf{R}^e\} = -\frac{1}{k_d} \int_{\Gamma_q} q [\mathbf{N}^e]^T d\Gamma_q + H_d \int_{\Gamma_h} T_{\infty} [\mathbf{N}^e]^T d\Gamma_h \quad (3.52)$$

onde $[\mathbf{N}^e]$ é a matriz formado pelas funções de forma em cada ponto nodal do elemento, $[\mathbf{B}^e]$ é a matriz das derivadas espaciais (ou laplaciano) e $[\mathbf{D}^e]$ é a matriz dos coeficientes do laplaciano, definidos como:

$$[\mathbf{N}^e] = \begin{bmatrix} N_1^e & N_2^e & N_3^e \end{bmatrix} \quad (3.53)$$

$$[\mathbf{B}^e] = \frac{1}{2A^e} \begin{bmatrix} b_1^e & b_2^e & b_3^e \\ c_1^e & c_2^e & c_3^e \end{bmatrix} \quad (3.54)$$

$$[\mathbf{D}^e] = \begin{bmatrix} k_r & 0 \\ 0 & k_z \end{bmatrix} \quad (3.55)$$

onde, adotando a hipótese de material isotrópico, $[\mathbf{D}^e] = k_d$.

Para um elemento triangular linear, as relações a seguir são válidas [18]:

$$\int_{\Omega} N_1^a N_2^b N_3^c d\Omega = \frac{a! b! c! 2A^e}{(a+b+c+2)!} \quad (3.56)$$

$$\int_{\Gamma} N_1^a N_2^b d\Gamma = \frac{a! b! l}{(a+b+1)!} \quad (3.57)$$

onde l é o comprimento do contorno da superfície. Essas relações são importantes para solução das integrais detalhadas anteriormente.

Aplicando as relações das Eqs. 3.53-3.57 nas Eqs. 3.50-3.52, obtemos:

$$[\mathbf{C}^e] = \frac{\pi \bar{r} A^e}{6\alpha_d} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \quad (3.58)$$

$$[\mathbf{K}^e] = 2\pi \bar{r} [\mathbf{B}^e]^T [\mathbf{B}^e] + \frac{H_d \pi l_{13}}{6} \begin{bmatrix} 3r_1 + r_3 & 0 & r_1 + r_3 \\ 0 & 0 & 0 \\ r_3 + r_1 & 0 & 3r_1 + r_3 \end{bmatrix} \quad (3.59)$$

$$\{\mathbf{R}^e\} = -\frac{\pi q l_{13}}{3k_d} \begin{bmatrix} 2r_1 + r_3 \\ 0 \\ r_1 + 2r_3 \end{bmatrix} + \frac{\pi H_d T_\infty l_{13}}{3} \begin{bmatrix} 2r_1 + r_3 \\ 0 \\ r_1 + 2r_3 \end{bmatrix} \quad (3.60)$$

onde \bar{r} é o raio médio do elemento definido como:

$$\bar{r} = \frac{r_1 + r_2 + r_3}{3} \quad (3.61)$$

Nas relações acima foi considerado que as forças térmicas foram aplicados no lado l_{13} do elemento triângular à título de representação. Caso o forçamento ocorra em outros lados do elemento, basta modificar a posição dos elementos nas matrizes e acompanhar a indexação de l_{ij} , como por exemplo:

$$\text{Forçamento no lado } l_{12}: \begin{bmatrix} 3r_1 + r_2 & r_1 + r_2 & 0 \\ r_1 + r_2 & 3r_1 + r_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{e} \quad \begin{bmatrix} 2r_1 + r_2 \\ r_1 + 2r_2 \\ 0 \end{bmatrix}$$

Capítulo 4

Algoritmo Computacional

Neste capítulo apresentaremos as principais características do código numérico desenvolvido na linguagem *Python* para o problema térmico no disco de freio considerando frenagem de emergência e múltiplas frenagens. Inicialmente é apresentado o *script* de geração da malha dos modelos para simulação. Posteriormente, a lógica de montagem das matrizes globais $[\mathbf{C}]$ e $[\mathbf{K}]$ no código são exibidos. Em seguida, apresentamos a aplicação das condições de contorno de *Neumann* e *Robin* e, por fim, o algoritmo de solução do sistema linear da equação de condução do calor do disco de freio.

4.1 Geração de Malha

O domínio consiste em uma malha estruturada com elementos triangulares (Figura 4.1) realizado através de um *script* em *Python*. Para auxiliar na aplicação dos vetores de força térmica, o domínio foi dividido em três subdomínios:

- **Malha 1:** $r_d \leq r < r_p \cup 0 \leq z \leq \delta_d$
- **Malha 2:** $r_p \leq r < R_p \cup 0 \leq z \leq \delta_d$
- **Malha 3:** $R_p \leq r \leq R_d \cup 0 \leq z \leq \delta_d$

A primeira etapa consistiu em particionar os subdomínios em n_r pontos nodais na direção radial e n_z pontos nodais na direção axial. Cada região foi formada por n_e elementos e n_p pontos nodais, onde procurou-se manter uma igualdade entre as áreas dos elementos através de uma ponderação na direção radial:

Algoritmo 1 *Script* de ponderação dos nós na direção radial das submalhas

Input: $r_d, r_p, R_p, R_d, n_{r2}$

Output: n_r

- 1: $n_{r1} = \text{int} \left(\frac{r_p - r_d}{R_p - r_p} ((n_{r2} - 1) + 1) \right)$ \triangleright N^o de nós na direção radial da malha 1
 - 2: $n_{r3} = \text{int} \left(\frac{R_d - R_p}{R_p - r_p} ((n_{r2} - 1) + 1) \right)$ \triangleright N^o de nós na direção radial da malha 3
 - 3: $n_r = n_{r1} + n_{r2} + n_{r3} - 2$ \triangleright N^o total de nós na direção radial
-

Esse processo foi utilizado com o objetivo de reduzir o tempo processamento da malha. Em seguida, iniciou-se o processo de geração da malha, como apresentado de forma genérica no *script* a seguir:

Algoritmo 2 *Script* de montagem da malha

Input: $r_d, r_p, R_p, R_d, n_{r2}, \dots, n_r$

Output: d_r, d_z, R, Z

- 1: $n_p = n_r n_z$ \triangleright N^o de nós da malha i
 - 2: $n_e = 2(n_r - 1)$ \triangleright N^o de elementos da malha i
 - 3: $L_r = (R_i - r_i)$ \triangleright Comprimento na direção r da malha i
 - 4: $L_z = \frac{\delta_p}{2}$ \triangleright Comprimento na direção z da malha i
 - 5: $zd = \text{np.linspace}(0, L_z, n_z)$ \triangleright Define a posição dos nós em z
 - 6: $rd = \text{np.linspace}(r_i, R_i, n_r)$ \triangleright Define a posição dos nós em r
 - 7: $Z, R = \text{np.meshgrid}(zd, rd)$ \triangleright Formação da malha i
 - 8: $Z = Z.\text{reshape}((n_z * n_r))$ \triangleright Reorganiza os pontos para plotagem
 - 9: $R = R.\text{reshape}((n_z * n_r))$ \triangleright Reorganiza os pontos para plotagem
 - 10: $d_r = \frac{L_r}{n_r - 1}$ \triangleright Espaçamento da malha i em r
 - 11: $d_z = \frac{L_z}{n_z - 1}$ \triangleright Espaçamento da malha i em z
-

Em seguida criou-se a *matriz de conectividade IEN* de cada malha. Adotou-se o sentido anti-horário para a formação dos elementos de modo a garantir que não tenha sobreposição de forçamentos em um elemento. O código numérico usado para criação de uma *IEN* genérica é apresentado abaixo:

Algoritmo 3 *Script* de geração da IEN

Input: n_r, n_z, n_e

Output: *IEN*

- 1: $IEN = \text{np.zeros}(n_e, 3)$ \triangleright Criação da matriz de conectividade
 - 2: $elemento = 0$ \triangleright Incremento representativo do elemento
-

```

3: for  $i \leftarrow 1, n_r$  do
4:   for  $j \leftarrow 1, n_z$  do
5:      $IEN[elemento] = [n_z j + i + 1, n_z j + i, n_z(j + 1) + i]$ 
6:      $IEN[elemento + 1] = [n_z(j + 1) + i, n_z(j + 1) + i + 1, n_z j + i + 1]$ 
7:      $elemento += 2$ 
8:   end for
9: end for

```

Por fim, somou-se as matrizes de conectividades de cada subdomínio, além dos números de pontos nodais n_p e elementos n_e para criação de uma malha global. A malha global é ilustrada na figura 4.1.

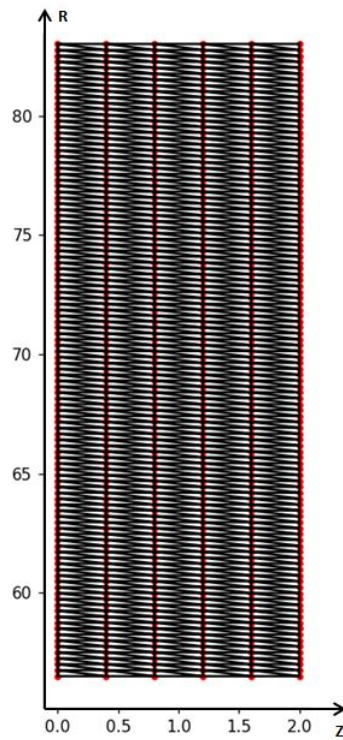


Figura 4.1: Malha bidimensional triangular gerada no *Python*

4.2 Montagem das Matrizes Globais

Após a criação da malha realizou-se a montagem das matrizes globais $[C]$ e $[K]$ e o vetor de força $\{R\}$. Novamente, o processo de montagem das matrizes foi realizada para cada subdomínio criado e, posteriormente, somadas para obter as matrizes globais do modelo.

Primeiramente, as áreas dos elementos A^e e a matriz Laplaciano $[\mathbf{B}^e]$ e sua transposta $[\mathbf{B}^e]^T$ foram calculados. Como o domínio foi dividido em elementos triangulares de lados dr e dz , temos:

$$A^e = \frac{1}{2} \begin{vmatrix} 1 & r_1 & z_1 \\ 1 & r_2 & z_2 \\ 1 & r_3 & z_3 \end{vmatrix} = \frac{dzdr}{2} \quad (4.1)$$

$$b_1^e = 0, \quad b_2^e = dz, \quad b_3^e = -dz \quad (4.2)$$

$$c_1^e = -dr, \quad c_2^e = dr, \quad c_3^e = 0$$

$$[\mathbf{B}^e] = \frac{1}{2A^e} \begin{bmatrix} 0 & dz & -dz \\ -dr & dr & 0 \end{bmatrix} \quad (4.3)$$

A transposta $[\mathbf{B}^e]^T$ foi obtida através da função `np.transpose()` da biblioteca `numpy` do `Python` e a multiplicação das matrizes $[\mathbf{B}^e]^T \cdot [\mathbf{B}^e]$ através da função `np.dot()`.

Posteriormente, obteve-se o vetor raio médio $\{\bar{r}\}$ através do código escrito a seguir:

Algoritmo 4 *Script* de geração do raio médio \bar{r}

Input: n_r, n_z, n_e, dr, r_d

Output: \bar{r}

- 1: $\bar{r} = \mathbf{np.zeros}(n_e, 1)$ ▷ Criação do vetor raio médio \bar{r}
 - 2: $elemento = 0$ ▷ Incremento representativo do elemento
 - 3: **for** $i \leftarrow 1, n_r$ **do**
 - 4: **for** $j \leftarrow 1, n_z$ **do**
 - 5: $\bar{r}[elemento] = (3r_d + (3j + 1)dr)/3$
 - 6: $\bar{r}[elemento + 1] = (3r_d + (3j + 2)dr)/3$
 - 7: $elemento + = 2$
 - 8: **end for**
 - 9: **end for**
-

Finalmente, utilizou-se o seguinte *script* para a montagem das matrizes globais $[K_c]$ e $[C]$:

Algoritmo 5 *Script* de montagem das matrizes $[K_c]$ e $[C]$

Input: $n_e, A^e, \alpha_d, [B], \dots, IEN$ **Output:** $[K_c]$ e $[C]$

```
1:  $k_c = 2\pi A^e [B]^T \cdot [B]$  ▷ Parcelas constantes das matrizes  $[K_c]$  e  $[C]$ 
2:  $c = \frac{\pi A^e}{6\alpha_d} \cdot [[2, 1, 1], [1, 2, 1], [1, 1, 2]]$ 
3: for  $elem \leftarrow 0, n_e$  do
4:   for  $i_{local} \leftarrow 0, 3$  do
5:      $i_{global} = IEN[elem, i_{local}]$ 
6:     for  $j_{local} \leftarrow 0, 3$  do
7:        $j_{global} = IEN[elem, j_{local}]$ 
8:        $K_c[i_{global}, j_{global}] = K_c[i_{global}, j_{global}] + \bar{r}[elem, 0] \cdot k_c[i_{local}, i_{local}]$ 
9:        $C[i_{global}, j_{global}] = C[i_{global}, j_{global}] + \bar{r}[elem, 0] \cdot c[i_{local}, j_{local}]$ 
10:    end for
11:  end for
12: end for
```

Como o coeficiente de transferência de calor por convecção h é função do tempo, a matriz de rigidez convectiva (ou matriz de contorno) $[K_h^e]$ é obtida dentro do *loop* do algoritmo de solução. Além disso, a matriz é avaliada no contorno, portanto se fez necessário localizar os pontos nodais no qual ocorre o a transferência de calor por convecção, como apresentado no código a seguir:

Algoritmo 6 *Script* de montagem da matriz de contorno $[K_h]$

Input: $n_e, n_z, H_d, dr, \bar{r}, \dots, IEN$ **Output:** $[K_h]$ ▷ Parcela constante da matriz $[K_h]$

```
1:  $k_h = \frac{H_d \pi dr}{6} \cdot [[4, 0, 2], [0, 0, 0], [2, 0, 4]]$ 
2:  $n_h = []$ 
3: for  $i \leftarrow n_{contorno}$  do
4:    $n_h = i$  ▷ Insere na lista  $n_h$  os nós na borda sob convecção
5: end for
```

```

6: for  $elem \leftarrow n_h$  do
7:   for  $i_{local} \leftarrow 0, 3$  do
8:      $i_{global} = IEN[elem, i_{local}]$ 
9:     for  $j_{local} \leftarrow 0, 3$  do
10:       $j_{global} = IEN[elem, j_{local}]$ 
11:       $K_h[i_{global}, j_{global}] = K_h[i_{global}, j_{global}] + \bar{r}[elem, 0] \cdot k_h[i_{local}, i_{local}]$ 
12:    end for
13:  end for
14: end for

```

No algoritmo de montagem da matriz de contorno $[K_h]$ utilizou-se a aproximação pelo raio médio \bar{r} com o objetivo de simplificar o processo de desenvolvimento do algoritmo:

$$[K_h] = \frac{H_d \pi l_{13}}{6} \begin{bmatrix} 3r_1 + r_3 & 0 & r_1 + r_3 \\ 0 & 0 & 0 \\ r_3 + r_1 & 0 & 3r_1 + r_3 \end{bmatrix} = \frac{H_d \pi \bar{r} l_{13}}{6} \begin{bmatrix} 4 & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & 4 \end{bmatrix} \quad (4.4)$$

4.3 Aplicação das Condições de Contorno

O modelo matemático envolve condições de contorno de *Neumann* e *Robin* (ou mista). A primeira não altera as matrizes globais do problema, enquanto que na segunda há a adição da matriz de contorno $[K_h]$, avaliada no contorno do modelo, nas matrizes globais. Além disso, os vetores de força são somados ao vetor solução $\{b\}$ na montagem do sistema linear. O procedimento ocorre no *loop* de solução do problema, pois a matriz precisa ser constantemente atualizada. O procedimento é demonstrado abaixo:

$$[K] = [K_c] + [K_h] \quad (4.5)$$

$$[A] = [K] + [C] \quad (4.6)$$

$$[A]\{x\} = \{b\} + \{R\}. \quad (4.7)$$

Como já citado, o fluxo de calor q e o coeficiente de transferência de calor por convecção h são dependentes do tempo, logo a formação do vetor de força $\{\mathbf{R}^e\}$ também é realizada dentro do *loop* do algoritmo de solução do sistema linear. Os algoritmos de montagem dos vetores de força térmica são apresentados abaixo:

Algoritmo 7 *Script* de montagem do vetor força térmica $\{R_h\}$

Input: $n_h, T_\infty, H_d, dr, \bar{r}, \dots, IEN$ **Output:** $\{R_h\}$ ▷ Parcela constante do vetor R_h

```
1:  $f_h = \frac{\pi H_d T_\infty dr}{3} \cdot [[3], [0], [3]]$ 
2: for  $elem \leftarrow n_h$  do
3:   for  $i_{local} \leftarrow 0, 3$  do
4:      $i_{global} = IEN[elem, i_{local}]$ 
5:      $F_h[i_{global}, 0] = F_h[i_{global}, 0] + \bar{r}[elem, 0] \cdot f_h[i_{local}, 0]$ 
6:   end for
7: end for
```

Algoritmo 8 *Script* de montagem do vetor força térmica $\{R_q\}$

Input: $\phi_0, k_d, P_{max}, \omega, dr, \bar{r}, \dots, IEN$ **Output:** $\{R_q\}$

```
1:  $q_0 = \frac{\phi_0 \mu P_{max} \omega \sigma}{2\pi k_d}$  ▷ Fluxo de calor inicial
2:  $f_q = \frac{\pi dr q_0}{3} \cdot [[3], [0], [3]]$  ▷ Parcela constante do vetor  $R_q$ 
3:  $n_q = []$ 
4: for  $i \leftarrow n_{contorno}$  do
5:    $n_q = i$  ▷ Insere na lista  $n_q$  os nós na borda sob fluxo de calor
6: end for
7: for  $elem \leftarrow n_q$  do
8:   for  $i_{local} \leftarrow 0, 3$  do
9:      $i_{global} = IEN[elem, i_{local}]$ 
10:    for  $j_{local} \leftarrow 0, 3$  do
11:       $j_{global} = IEN[elem, j_{local}]$ 
12:       $F_q[i_{global}, j_{global}] = F_q[i_{global}, j_{global}] + \bar{r}[elem, 0] \cdot f_q[i_{local}, i_{local}] \cdot R[i_{global}]$ 
13:    end for
14:  end for
15: end for
```

No algoritmo de montagem dos vetores de força térmica $\{R_h\}$ e $\{R_q\}$ utilizou-se a aproximação pelo raio médio \bar{r} com o objetivo de simplificar o processo de

desenvolvimento do algoritmo:

$$\{\mathbf{R}_h\} = \frac{\pi H_d T_\infty l_{13}}{3} \begin{bmatrix} 2r_1 + r_3 \\ 0 \\ r_1 + 2r_3 \end{bmatrix} = \frac{\pi \bar{r} H_d T_\infty l_{13}}{3} \begin{bmatrix} 3 \\ 0 \\ 3 \end{bmatrix} \quad (4.8)$$

$$\{\mathbf{R}_q\} = -\frac{\pi q l_{13}}{3k_d} \begin{bmatrix} 2r_1 + r_3 \\ 0 \\ r_1 + 2r_3 \end{bmatrix} = -\frac{\pi \bar{r} q l_{13}}{3k_d} \begin{bmatrix} 3 \\ 0 \\ 3 \end{bmatrix} \quad (4.9)$$

4.4 Algoritmo de Solução

Nessa seção descrevemos, separadamente, o passo-a-passo do algoritmo de solução para o modelo de frenagem única (ou de emergência) e para o modelo de múltiplas frenagens.

4.4.1 Frenagem de Emergência

Após a criação da malha, montagem das matrizes globais e aplicação das condições de contorno, elaborou-se o algoritmo de solução da seguinte forma:

1. Definiu-se o espaço de tempo dt para simulação;
2. Inicia-se um *loop* que particiona o tempo de frenagem t_b de acordo com o espaço de tempo dt ;
3. A cada instante, o tempo de frenagem e, conseqüentemente, a velocidade angular do veículo são calculados;
4. Calcula-se os vetores de força devido ao fluxo de calor $\{R_q\}$ e devido à transferência de calor por convecção $\{R_h\}$. Simultaneamente, calcula-se a matriz de contorno $[K_h^e]$.
5. Realiza-se a montagem do sistema linear:

$$[A] = \frac{[C]}{dt} + \theta[K] \quad (4.10)$$

$$[B] = \frac{[C]}{dt} - (1 - \theta)[K] \quad (4.11)$$

onde $[K] = [K_c] + [K_h]$.

6. O vetor $\{b\}$ do sistema linear é calculado a partir da inversa da matriz $[B]$ e dos vetores de força $\{R_q\}$ e $\{R_h\}$, ou seja:

$$\{b\} = [B]^{-1}\{T\}_n + \{R_h\} - \{R_q\} \quad (4.12)$$

7. Por fim, o sistema linear é calculado pelo método de decomposição L.U. através da função `np.linalg.solve()` da biblioteca *Numpy* e a simulação se encerra quando o tempo de frenagem t_b for atingido.

A montagem do sistema linear e o método de resolução é mostrado abaixo:

Algoritmo 9 *Script* de montagem do sistema linear

Input: $[C], [K], dt, \dots, \theta$

Output: T

- | | |
|--|--------------------------------------|
| 1: $[A] = \frac{[C]}{dt} + \theta[K]$ | ▷ Matriz que compões o tempo $n + 1$ |
| 2: $[B] = \frac{[C]}{dt} - (1 - \theta)[K]$ | ▷ Matriz que compões o tempo n |
| 3: $\{b\} = [B] \cdot \{T\} + \{R_h\} - \{R_q\}$ | |
| 4: $\{T\} = \mathbf{np.linalg.solve}(A, b)$ | ▷ Solução numérica |
-

4.4.2 Múltiplas Frenagens

O método de múltiplas frenagens utiliza o algoritmo do método anterior em seu algoritmo, contudo há o acréscimo de uma etapa em que o veículo está acelerando e resfriando o disco de freio por convecção. Logo, elaborou-se o algoritmo da seguinte forma:

1. Definiu-se o espaço de tempo dt para simulação, o número de frenagens n_b e o tempo de resfriamento Δt ;
2. Inicia-se um *loop* que caracteriza a quantidade de frenagens definida previamente;
3. Inicia-se um outro *loop* que particiona o tempo total da simulação pelo espaço de tempo dt ;
4. A cada instante, o tempo de frenagem e, conseqüentemente, a velocidade angular do veículo é calculada;
5. Cria-se a partição do fenômeno com funções condicionais *if-else*, ou seja:

```

if tempo = tb:
    Resolve o loop de resfriamento do
    disco
else :
    Resolve o loop do método de frenagem
    de emergência

```

6. Calcula-se os vetores de força devido ao fluxo de calor $\{R_q\}$ e devido à transferência de calor por convecção $\{R_h\}$. Simultaneamente, calcula-se a matriz de contorno $[K_h^e]$.

7. Realiza-se a montagem do sistema linear:

$$[A] = \frac{[C]}{dt} + \theta[K] \quad (4.13)$$

$$[B] = \frac{[C]}{dt} - (1 - \theta)[K] \quad (4.14)$$

onde $[K] = [K_c] + [K_h]$.

8. O vetor $\{b\}$ do sistema linear é calculado a partir da inversa da matriz $[B]$ e dos vetores de força $\{R_q\}$ e $\{R_h\}$, ou seja:

$$\{b\} = [B]^{-1}\{T\}_n + \{R_h\} - \{R_q\} \quad (4.15)$$

9. Por fim, o sistema linear é calculado e a simulação se encerra quando o tempo de frenagem t_b somado ao tempo de resfriamento Δt para o número de frenagens n_b especificado for atingido.

Capítulo 5

Validação

5.1 Condução de Calor Transiente em um Cilindro Curto

Devido a ausência de soluções analíticas que representem de forma precisa os modelos de frenagens descritos nos capítulos anteriores, surge a necessidade de validar o modelo numérico desenvolvido através de um modelo clássico representativo que possua solução analítica.

O modelo clássico utilizado para validação consiste um problema transiente axisimétrico de um cilindro curto com altura e comprimentos iguais sujeito à efeitos convectivos na extremidade externa e isolado nas extremidades interna e em suas laterais. Nessa seção será desenvolvido o modelo físico e numérico do problema, bem como os resultados do problema.

5.1.1 Modelo Teórico

Conforme citado anteriormente, o problema transiente axissimétrico de um cilindro curto sob efeitos convectivos em sua extremidade externa torna-se um caso representativo para validação dos modelos apresentados nesse trabalho pelos seguintes motivos:

1. A equação que governa a condução de calor no cilindro, no modelo numérico, é igual à equação do de condução do calor no disco de freio, representado pela Eq. 3.19;

2. O modelo clássico possui geometria semelhante e configurações iguais ao modelo térmico do disco de freio, ou seja, axissimétrico e transiente;
3. O problema possui solução analítica baseada nas cartas de temperaturas unidimensionais da condução de calor em regime transiente para cilindros [5], uma vez que o domínio do problema e as condições de contorno permitem a hipótese de condução de calor unidimensional;
4. As técnicas de discretização do domínio espacial e discretização do tempo empregadas no modelo numérico são as mesmas utilizadas no modelo térmico do disco de freio.

Para formulação do modelo físico adotou-se as seguintes hipóteses:

- As propriedades do material são constantes e independente da temperatura ao longo de uma seção, ou seja, considera-se um material isotrópico;
- O calor é dissipado somente para o ambiente pela extremidade externa do cilindro convecção, ou seja, a dissipação de calor por radiação é negligenciada;
- A temperatura do fluido T_∞ é constante;
- O coeficiente de transferência de calor por convecção h é constante, pois o cilindro se mantém estático.

A Fig. 5.1 ilustra o modelo descrito nessa seção e suas condições de contorno. O problema transiente analisado desenvolve-se em um cilindro de altura e comprimento iguais isolado em suas extremidades laterais e interna, sujeito à efeitos convectivos em sua borda externa. O cilindro inicialmente é mantido isolado à uma temperatura inicial T_0 e, subitamente é colocado em contato com um ambiente à temperatura T_∞ , dissipando calor por convecção.

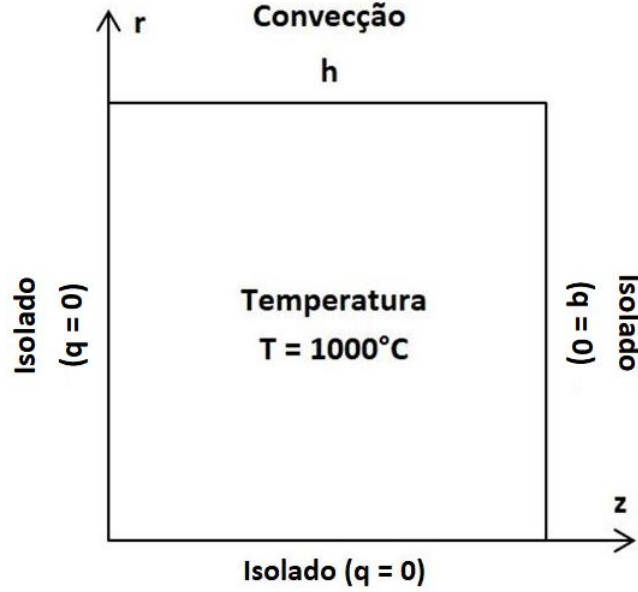


Figura 5.1: Condições de contorno no cilindro curto

Para descrever o fenômeno térmico no cilindro foi considerado a equação do calor apresentada na equação 3.19, porém em sua forma adimensional [23]. A formulação adimensional torna-se necessário pois as cartas de temperaturas que apresentam uma solução analítica para o problema térmico estão em função dos parâmetros adimensionais definidos abaixo:

$$\text{Número de Biot: } Bi = \frac{hR}{k} \quad (5.1)$$

$$\text{Número de Fourier: } \tau = \frac{\alpha t}{R^2} \quad (5.2)$$

$$\text{Coordenada radial adimensional: } \bar{r} = \frac{r}{R} \quad (5.3)$$

$$\text{Coordenada axial adimensional: } \bar{z} = \frac{z}{L} \quad (5.4)$$

$$\text{Temperatura adimensional: } \theta_c = \frac{T - T_\infty}{T_0 - T_\infty} \quad (5.5)$$

onde R é o raio do cilindro e L é o comprimento do cilindro. Dessa forma, a equação do calor adimensional em coordenadas cilíndricas para o cilindro axissimétrico transiente é definida como:

$$\frac{\partial^2 \theta_c}{\partial \bar{r}^2} + \frac{1}{\bar{r}} \frac{\partial \theta_c}{\partial \bar{r}} + \frac{\partial^2 \theta_c}{\partial \bar{z}^2} = \frac{\partial \theta_c}{\partial \tau}, \quad (5.6)$$

$$0 \leq \bar{r} \leq 1, \quad 0 \leq \bar{z} \leq 1 \quad e \quad \tau \geq 0$$

com as seguintes condições de contorno:

$$\left. \frac{\partial \theta_c}{\partial \bar{z}} \right|_{\bar{z}=0} = 0 \quad em \ 0 \leq \bar{r} \leq 1 \ e \ \tau \geq 0 \quad (5.7)$$

$$\left. \frac{\partial \theta_c}{\partial \bar{z}} \right|_{\bar{z}=1} = 0 \quad em \ 0 \leq \bar{r} \leq 1 \ e \ \tau \geq 0 \quad (5.8)$$

$$\left. \frac{\partial \theta_c}{\partial \bar{r}} \right|_{\bar{r}=0} = 0 \quad em \ 0 \leq \bar{z} \leq 1 \ e \ \tau \geq 0 \quad (5.9)$$

$$\left. \frac{\partial \theta_c}{\partial \bar{r}} \right|_{\bar{r}=1} + Bi\theta_c = 0 \quad em \ 0 \leq \bar{z} \leq 1 \ e \ \tau \geq 0 \quad (5.10)$$

$$\theta_c = 1 \quad em \ 0 \leq \bar{r} \leq 1 \cup 0 \leq \bar{z} \leq 1 \ e \ \tau = 0 \quad (5.11)$$

As condições de contorno e a geometria do problema sugerem que o problema possa ser reduzido para uma formulação unidimensional transiente em coordenadas cilíndricas. Portanto, para a solução analítica, as equações 5.6-5.10 se reduzem à seguinte formulação:

$$\frac{1}{\bar{r}} \frac{\partial \theta_c}{\partial \bar{r}} \left(\bar{r} \frac{\partial \theta_c}{\partial \bar{r}} \right) = \frac{\partial \theta_c}{\partial \tau} \quad em \ 0 \leq \bar{r} \leq 1 \ e \ \tau \geq 0 \quad (5.12)$$

$$\left. \frac{\partial \theta_c}{\partial \bar{r}} \right|_{\bar{r}=0} = 0 \quad para \ \tau \geq 0 \quad (5.13)$$

$$\left. \frac{\partial \theta_c}{\partial \bar{r}} \right|_{\bar{r}=1} + Bi\theta_c = 0 \quad para \ \tau \geq 0 \quad (5.14)$$

$$\theta_c = 1 \quad em \ 0 \leq \bar{r} \leq 1 \ e \ \tau = 0 \quad (5.15)$$

A formulação acima possui solução analítica exata quando o número de Biot é nulo (sistemas concentrados). Contudo, para valores de $Bi \neq 0$ a solução analítica é aproximada por séries infinitas que convergem rapidamente com o aumento de tempo e para tempos $t \geq 0.2s$ [23]. Os valores comumente são apresentados em formas gráficas, no entanto, nesse trabalho calcularemos os valores a partir da solução analítica aproximada para condução de calor unidimensional no cilindro:

$$\theta_c(\bar{r}, \tau) = \frac{T(r, t) - T_\infty}{T_0 - T_\infty} = A_1 e^{-\lambda_1^2 \tau} J_0(\lambda_1 r/R), \quad \tau \geq 0.2 \quad (5.16)$$

$$\text{Função de Bessel de ordem 0: } J_0(\lambda_1 r/R) = \sum_{k=0}^{\infty} \frac{(-1)^k}{k! k!} \left(\frac{\lambda_1 r}{2R} \right)^{2k} \quad (5.17)$$

onde A_1 e λ_1 são constantes que dependem do número de Biot. Para obter valores da solução analítica aproximada desenvolveu-se um código em *Python* apresentado no Apêndice A.1.

5.1.2 Modelo Numérico

Para desenvolvimento do modelo numérico, utilizou-se a formulação do problema representado pelas equações gerais:

$$\frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \frac{\partial T}{\partial r} + \frac{\partial^2 T}{\partial z^2} = \frac{1}{\alpha} \frac{\partial T}{\partial t}, \quad 0 \leq r \leq R, \quad 0 \leq z \leq L, \quad t > 0 \quad (5.18)$$

$$\left. \frac{\partial T}{\partial z} \right|_{z=0} = 0 \quad \text{em } 0 \leq r \leq R \text{ e } t \geq 0 \quad (5.19)$$

$$\left. \frac{\partial T}{\partial z} \right|_{z=L} = 0 \quad \text{em } 0 \leq r \leq R \text{ e } t \geq 0 \quad (5.20)$$

$$\left. \frac{\partial T}{\partial r} \right|_{r=0} = 0 \quad \text{em } 0 \leq z \leq L \text{ e } t \geq 0 \quad (5.21)$$

$$-\left. \frac{\partial T}{\partial r} \right|_{r=R} = H(T_\infty - T) \quad \text{em } 0 \leq z \leq L \text{ e } t \geq 0 \quad (5.22)$$

$$T = T_0 \quad \text{em } 0 \leq r \leq R \cup 0 \leq z \leq L \text{ e } t = 0 \quad (5.23)$$

onde $H = h/k$.

Análogo ao procedimento utilizado para o problema térmico do disco de freio no Capítulo 3 deste trabalho, aplicou-se o Método de Elementos Finitos ao modelo em questão. Empregou-se o Método de Resíduos Ponderados na equação de condução do calor do problema e o Teorema de Green para obter a formulação fraca da equação, descrita abaixo:

$$\begin{aligned} & - \int_{\Omega} \left[\frac{\partial \omega}{\partial r} \frac{\partial T}{\partial r} + \frac{\partial \omega}{\partial z} \frac{\partial T}{\partial z} - \frac{\omega}{r} \frac{\partial T}{\partial r} \right] d\Omega + H \int_{\Gamma_h} \omega T d\Gamma_h \\ & + \frac{1}{\alpha} \int_{\Omega} \omega \frac{\partial T}{\partial t} d\Omega + H \int_{\Gamma_h} \omega T_\infty d\Gamma_h = 0 \end{aligned} \quad (5.24)$$

A discretização espacial de Galerkin e a discretização temporal por diferenças finitas também foram adotadas. O modelo em sua forma matricial totalmente discretizada é apresentada pela Eq. 3.44, porém o vetor força térmica não possui o termo de fluxo de calor conforme apresentado no modelo do disco de freio.

A malha utilizada consistiu na mesma malha gerada do modelo do disco de freio, utilizando elementos triangulares lineares axissimétricos, cuja as funções de forma estão definidas nas Eqs. 3.45-3.47. Seguindo o mesmo procedimento adotado para a obter a forma matricial das equações, obtemos as matrizes globais e o vetor

de força do problema:

$$[\mathbf{C}^e] = \frac{\pi \bar{r} A^e}{6\alpha} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \quad (5.25)$$

$$[\mathbf{K}^e] = 2\pi \bar{r} [\mathbf{B}^e][\mathbf{B}^e]^T + \frac{H\pi l_{12}}{6} \begin{bmatrix} 3r_1 + r_2 & r_1 + r_2 & 0 \\ r_1 + r_2 & 3r_1 + r_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (5.26)$$

$$\{\mathbf{R}^e\} = \frac{\pi H T_\infty l_{12}}{3} \begin{bmatrix} 2r_1 + r_2 \\ r_1 + 2r_2 \\ 0 \end{bmatrix} \quad (5.27)$$

Desenvolveu-se um algoritmo (Apêndice A.2) elaborado em *Python* para obter a solução do problema utilizando procedimentos análogos apresentados no Capítulo 4. Com os valores analíticos e numéricos da distribuição de temperatura no cilindro podemos verificar a validade do modelo numérico do disco de freio, objetivo da seção a seguir.

5.1.3 Resultados

Com o objetivo de validar o código numérico do modelo térmico do disco de freio, realizou-se simulações do modelo do cilindro transiente e os resultados obtidos foram comparados com a solução analítica aproximada. As simulações realizadas foram:

1. Simulações fixando o número de Biot e variando o número de Fourier:

$$Bi = 1.0 \text{ e } \tau = \begin{cases} 0.4 \\ 0.8 \\ 1.2 \end{cases}$$

2. Simulações fixando o número de Fourier e variando o número de Biot:

$$\tau = 0.4 \text{ e } Bi = \begin{cases} 0.01 \\ 0.1 \\ 1.0 \end{cases}$$

A tabela 5.1 apresenta os dados de entrada utilizados nas simulações:

Raio interno do cilindro (r_0)	0 (m)
Raio externo do cilindro (R)	1 (m)
Comprimento do cilindro (L)	1 (m)
Condutividade térmica do material (k)	1 (W/m°C)
Calor específico do material (c)	1 (J/kg°C)
Densidade do material (ρ)	1 (kg/m ³)
Temperatura inicial do cilindro (T_0)	1000 (°C)
Temperatura do fluido externo (T_∞)	0°C

Tabela 5.1: Parâmetros e propriedades constantes utilizados nas simulações

Para as primeiras simulações na qual o número de Biot é fixado em $Bi = 1.0$, o coeficiente de transferência de calor por convecção foi definido como $h = 1\text{W}/\text{m}^2\text{C}$, pois:

$$Bi = \frac{hR}{k} = 1 \quad (5.28)$$

Para simular a variação do número de Fourier, variou-se o tempo de troca de calor entre o cilindro e o fluido, pois:

$$\alpha = \frac{k}{\rho c} = 1 \frac{\text{m}^2}{\text{s}} \quad (5.29)$$

$$t = \begin{cases} 0.4 \\ 0.8 \\ 1.2 \end{cases} \quad \therefore \tau = \frac{\alpha t}{R^2} = \begin{cases} 0.4 \\ 0.8 \\ 1.2 \end{cases} \quad (5.30)$$

Os valores das constantes A_1 e λ_1 da solução analítica aproximada para $Bi = 1.0$ são:

$$A_1 = 1.2071 \quad e \quad \lambda = 1.2558 \quad (5.31)$$

Em relação ao modelo numérico, a Tabela 5.2 apresenta as informações gerais utilizadas na simulação.

Para determinação do número de nós que seriam utilizados na simulação, realizou-se uma análise de convergência de malha adotando $Bi = 1.0$ e $\tau = 0.4$ como ilustrado na Figura 5.2. Verifica-se os valores convergem e estabilizam conforme o número de nós se aproxima de 625 unidades.

Número de nós (n_p)	625 (und.)
Número de elementos (n_e)	1152 (und.)
Espaço de tempo (dt)	0.001 (s)
Discretização temporal	Crank-Nicolson

Tabela 5.2: Parâmetros numéricos utilizados nas simulações

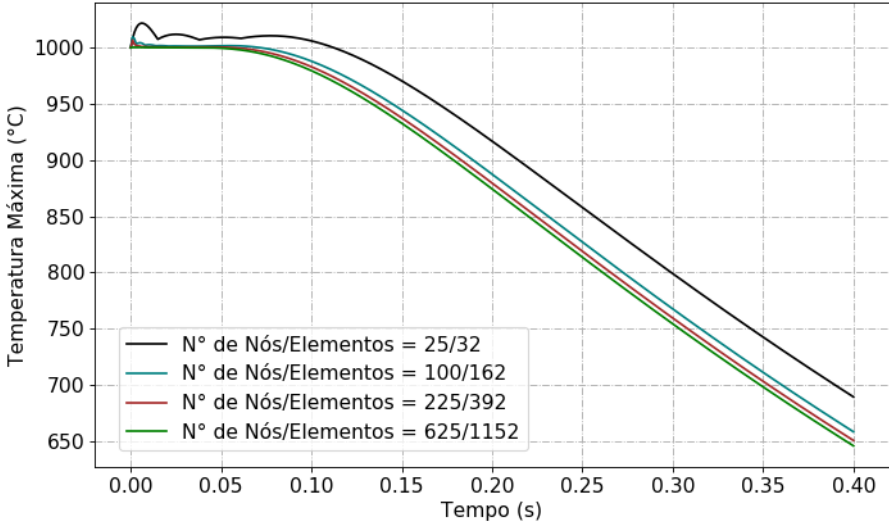
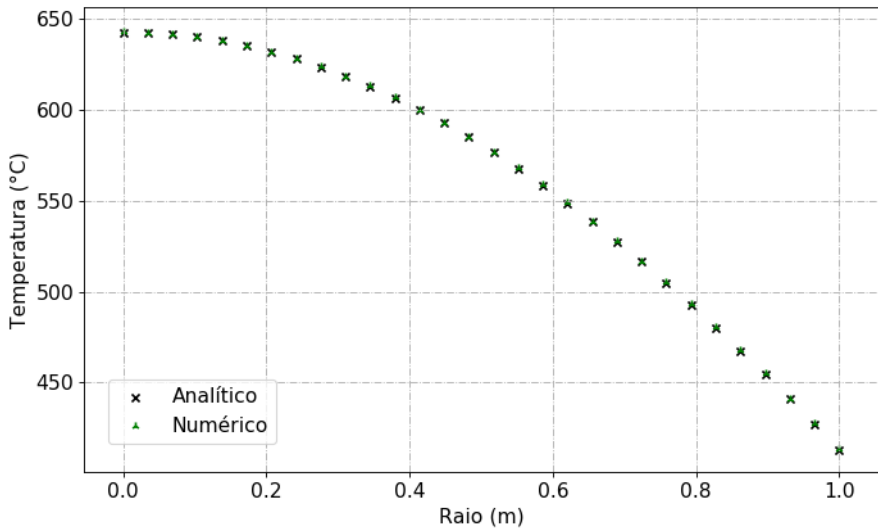
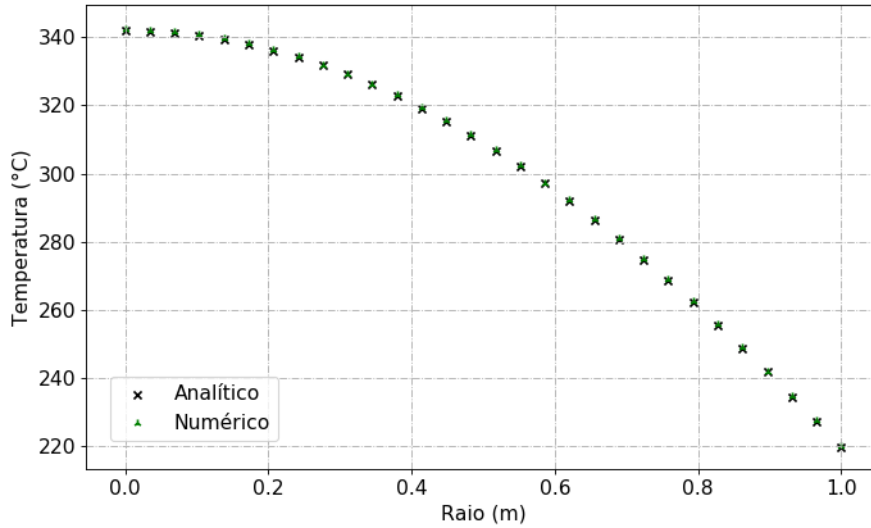


Figura 5.2: Análise de convergência de malha considerando $Bi = 1.0$ e $\tau = 0.4$

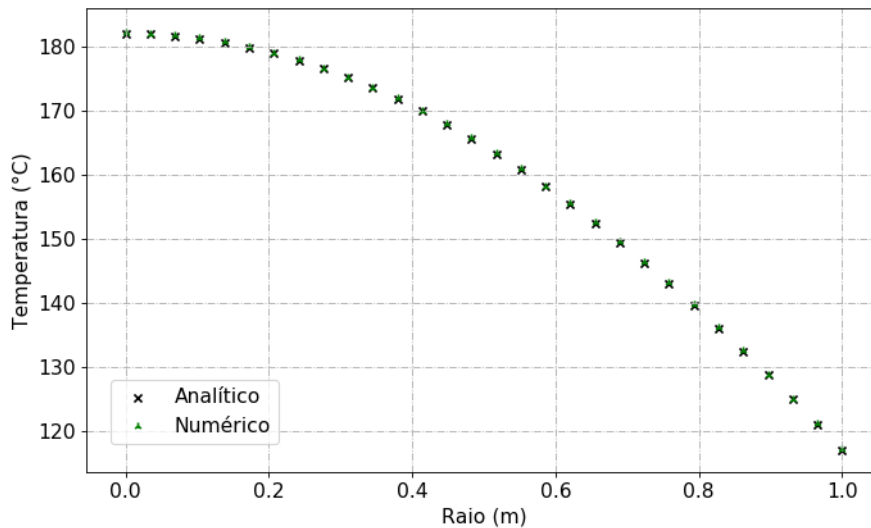
Os resultados comparativos são mostrados a seguir:



(a)



(b)



(c)

Figura 5.3: Gráficos comparativos entre as soluções analíticas e numéricas para (a) $\tau = 0.4$, (b) $\tau = 0.8$ e (c) $\tau = 1.2$ considerando $Bi = 1.0$

De acordo com os resultados, os valores numéricos obtidos próximo do centro do cilindro ($r = 0$) e na extremidade ($r = R$) possuíram as maiores discrepâncias em relação à solução analítica, como ilustrado Figura 5.4. Este comportamento já era esperado pois a utilização do raio médio \bar{r} do elemento na montagem das matrizes globais e vetores de força é um fator de geração de erros muito pronunciados.

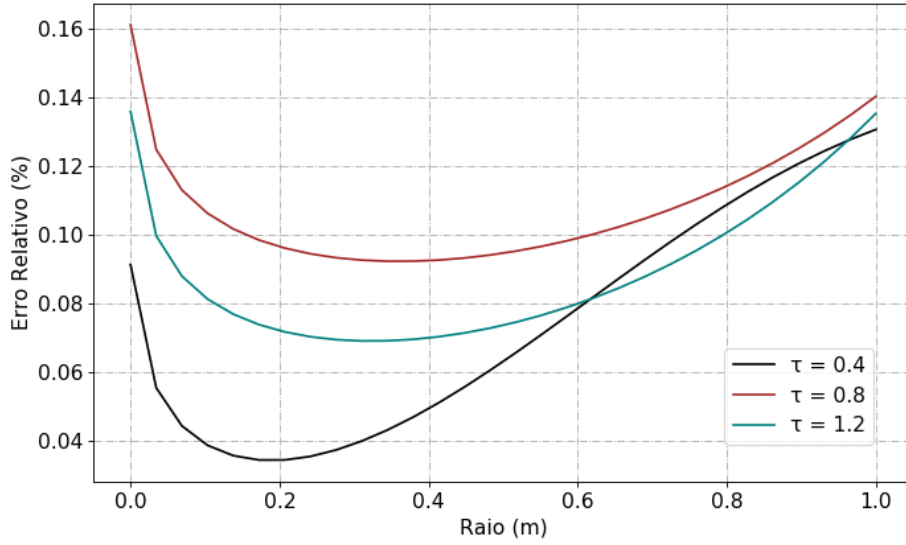


Figura 5.4: Erro relativo entre as soluções analítica e numérica para $Bi = 1.0$

A Tabela 5.3 ilustra os erros máximo e mínimos obtidos nas simulações.

Número de Fourier (τ)	0.4	0.8	1.2
Erro Relativo Mínimo (%)	0.035	0.09	0.07
Erro Relativo Máximo (%)	0.09	0.16	0.135

Tabela 5.3: Erros relativos mínimos e máximos considerando $Bi = 1.0$

De forma análoga às simulações anteriores, realizou-se simulações tendo como fator fixo o Número de Fourier como $\tau = 0.4$. O número de Biot foi variado através do coeficiente de transferência de calor por convecção h , ou seja:

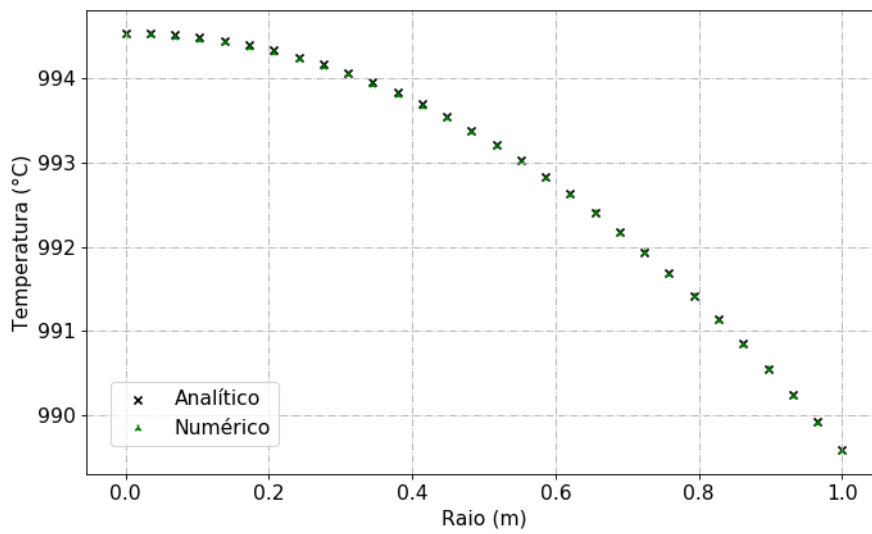
$$h = \begin{cases} 0.01 \\ 0.1 \\ 1.0 \end{cases} \frac{W}{m^2 \cdot ^\circ C} \quad \therefore \quad Bi = \frac{hR}{k} = \begin{cases} 0.01 \\ 0.1 \\ 1.0 \end{cases} \quad (5.32)$$

Os valores das constantes A_1 e λ_1 da solução analítica aproximada para os valores Bi definidos anteriormente são apresentados na Tabela 5.4.

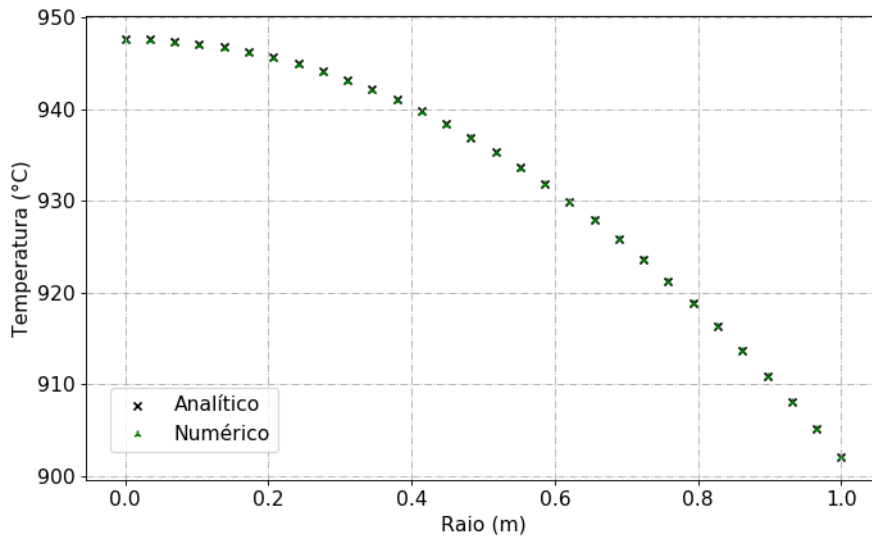
Número de Biot (Bi)	0.01	0.1	1.0
Constante A_1	1.0025	1.0246	1.2071
Constante λ_1	0.1412	0.4417	1.2558

Tabela 5.4: Constantes A_1 e λ_1 da solução analítica para os valores de Bi especificados

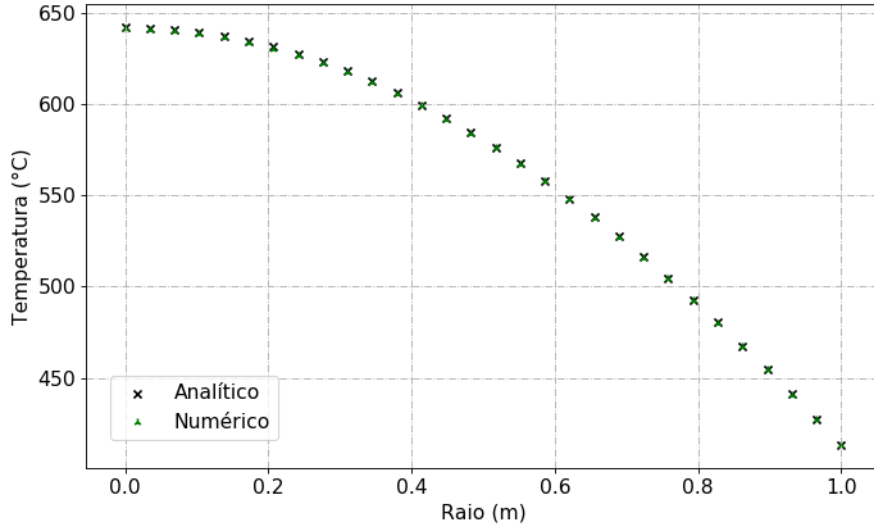
Os mesmos parâmetros numéricos apresentados na Tabela 5.2 foram utilizados nas simulações em questão. Os resultados comparativos são mostrados a seguir.



(a)



(b)



(c)

Figura 5.5: Gráficos comparativos entre as soluções analítica e numérica para (a) $Bi = 0.01$, (b) $Bi = 0.1$ e (c) $Bi = 1.0$ considerando $\tau = 0.4$

De acordo com os resultados, os valores numéricos obtidos próximo do centro do cilindro ($r = 0$) possuíram as maiores discrepâncias em relação à solução analítica, como ilustrado Figura 5.6. Este comportamento já era esperado pois a utilização do raio médio \bar{r} do elemento na montagem das matrizes globais e vetores de força é um fator de geração de erros muito pronunciados. Verifica-se que para valores de Bi próximo de zero, o erro relativo em relação à solução analítica é praticamente nulo, pois o o problema térmico se aproxima ao modelo de sistemas concentrados.

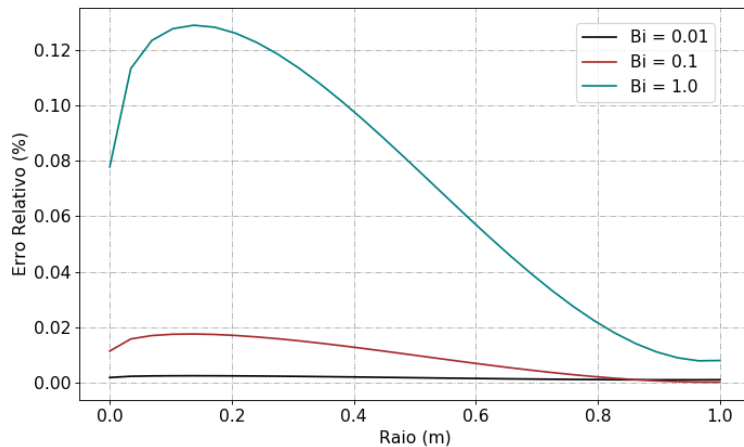


Figura 5.6: Erro relativo entre as soluções analítica e numérica para $\tau=0.4$

A Tabela 5.5 ilustra os erros máximo e mínimos obtidos nas simulações.

Número de Biot (Bi)	0.01	0.1	1.0
Erro Relativo Mínimo (%)	0.0	0.0	0.01
Erro Relativo Máximo (%)	0.0	0.01	0.13

Tabela 5.5: Erros relativos mínimos e máximos considerando $\tau = 0.4$

Para fins ilustrativos, a Figura 5.7 ilustra a simulação considerando $Bi = 1.0$ e $\tau = 0.4$ em quatro instantes de tempos durante o tempo total $t = 0.4s$.

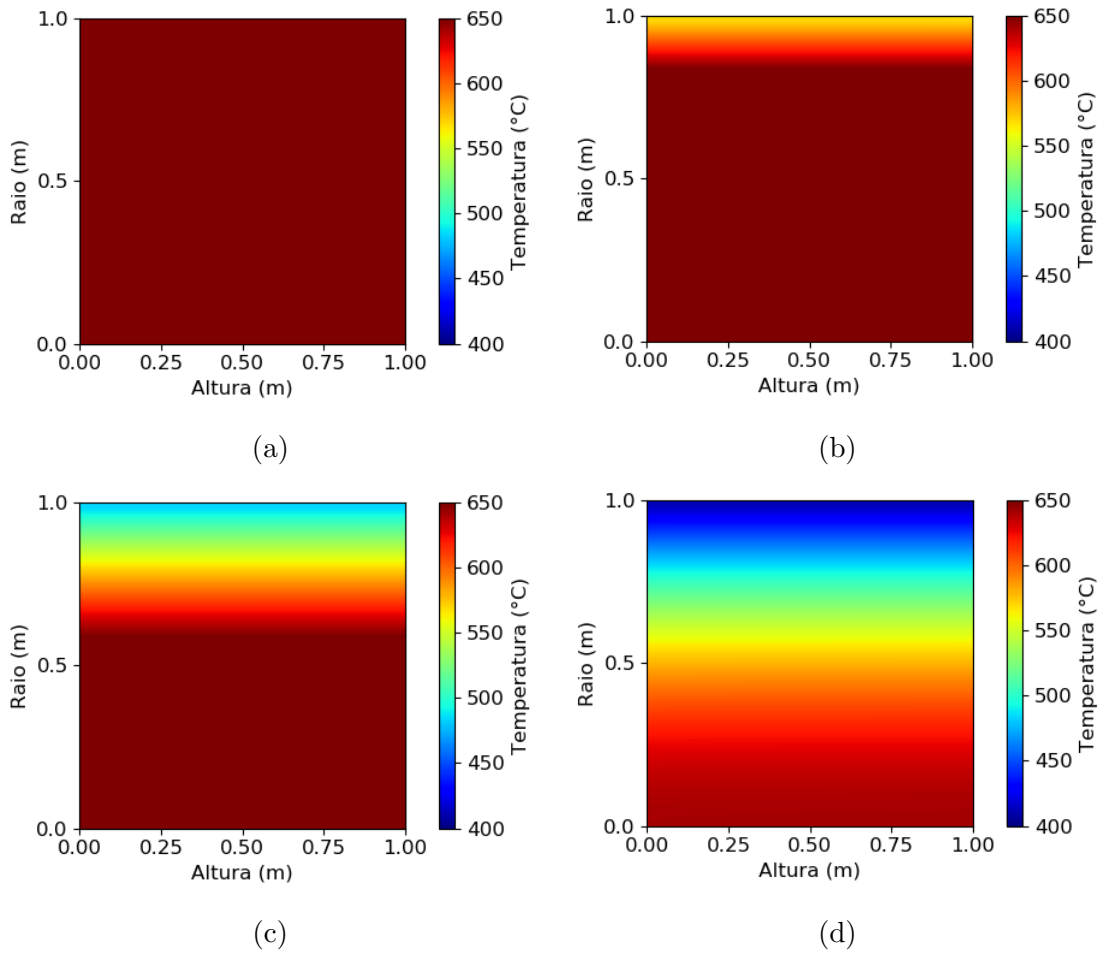


Figura 5.7: Simulação numérica do cilindro nos instantes (a) $t = 0.1s$, (b) $t = 0.2s$, (c) $t = 0.3s$ e (d) $t = 0.4s$ considerando $Bi = 1.0$ e $\tau = 0.4$

Considerando a deficiência inerente ao processo de resolução que utiliza o raio médio \bar{r} , os resultados obtidos permitem concluir que o sistema desenvolvido para domínios bidimensionais axissimétrico funciona satisfatoriamente com erros relativos abaixo de 1%.

Capítulo 6

Resultados

Nesse capítulo serão mostrados os resultados obtidos pela simulação numérica utilizando o Método de Elementos Finitos considerando os modelos de frenagem de emergência e múltiplas frenagens. Os dados de entrada para as simulações foram fornecidos pela Equipe Minerva Baja da Universidade Federal do Rio de Janeiro referentes ao projeto do protótipo de 2019, ilustrado na Figura 6.1 abaixo.



Figura 6.1: Veículo Baja SAE da Equipe Minerva Baja

6.1 Frenagem de Emergência

As geometrias simplificadas do disco de freio e da pastilha utilizados nas simulações numéricas é ilustrado na Figura 6.2. Nela são listados os principais parâmetros geométricos e resumidos na Tabela 6.1.

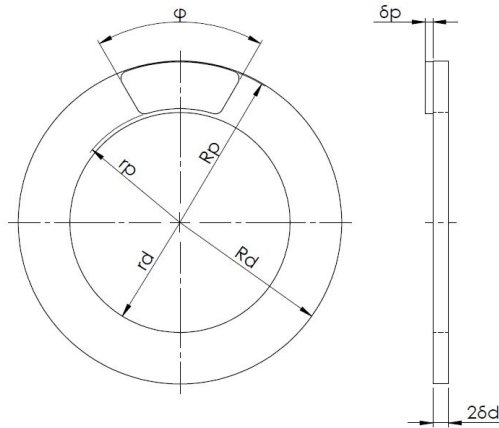


Figura 6.2: Geometria simplificada do mecanismo disco-pastilha de freio e seus principais parâmetros

Raio interno do disco (r_d)	56.5	mm
Raio interno da pastilha (r_p)	58.5	mm
Raio externo da pastilha (R_p)	82.5	mm
Raio externo do disco (R_d)	83.0	mm
Espessura do disco de freio ($2\delta_d$)	4	mm
Espessura da pastilha (δ_p)	3	mm
Ângulo de contato da pastilha (ϕ_0)	60	deg

Tabela 6.1: Parâmetros geométricos utilizados na simulações numéricas

As propriedades dos materiais do mecanismo disco-pastilha de freio e do fluido (ar) são ilustradas na Tabela 6.2 e Tabela 6.3, respectivamente. Os parâmetros de projeto fornecidos pela equipe Minerva Baja e adotados nas simulações são apresentados na Tabela 6.4.

Parâmetros termo-físicos	Disco	Pastilha
Condutividade térmica ($\text{W/m}\cdot\text{C}$)	42	12
Densidade (kg/m^3)	7200	2500
Capacidade térmica ($\text{J/kg}\cdot\text{C}$)	469	900

Tabela 6.2: Propriedades dos materiais utilizados na simulações numéricas

Temperatura do fluido (T_∞)	25	°C
Condutividade térmica (k_a)	0.0243	W/m°C
Viscosidade cinemática (ν_a)	1.0×10^{-6}	m/s ²

Tabela 6.3: Parâmetros do fluido (ar) utilizados na simulações numéricas

Coefficiente de atrito disco-pastilha (μ)	0.40	
Temperatura inicial do disco (T_0)	30	°C
Pressão do sistema (p)	3.64	MPa
Velocidade inicial do veículo (v_0)	15	m/s
Tempo de frenagem (t_b)	2.65	s
Raio efetivo da roda (R_r)	267	mm

Tabela 6.4: Parâmetros de projeto utilizados na simulações numéricas

Após a definição dos dados de entrada, definiu-se os parâmetros característicos da simulação numérica. A Tabela 6.5 resume os parâmetros utilizados na simulação.

Número de nós (n_p)	648 (und.)
Número de elementos (n_e)	1070 (und.)
Espaço de tempo (dt)	0.001 (s)
Discretização temporal	Crank-Nicolson

Tabela 6.5: Parâmetros numéricos utilizados na simulação de frenagem de emergência

O número de nós n_p e número de elementos n_e foram definidos a partir de uma análise de convergência de malha adotando como parâmetros o tempo de frenagem e a temperatura máxima no disco de freio.

Como ilustrado na Figura 6.3, a convergência dos valores da temperatura máxima no disco de freio durante o tempo de frenagem ocorre a partir de $n_p = 648$ und., justificando a escolha apresentada na Tabela 6.5.

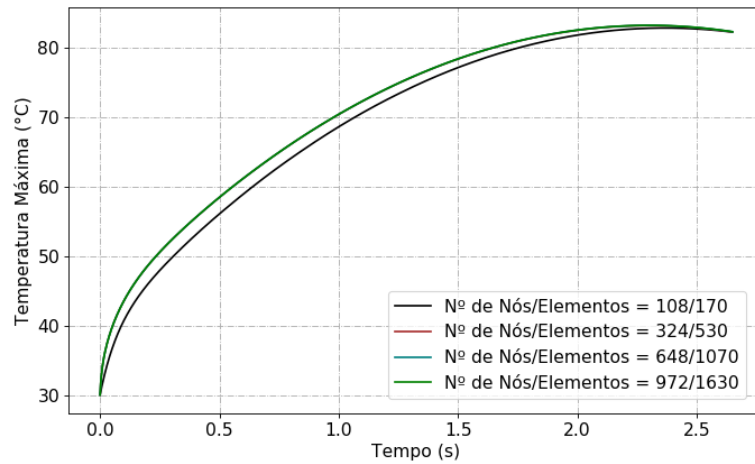


Figura 6.3: Análise de convergência de malha para simulação de frenagem de emergência

A Figura 6.4 tem o objetivo de ilustrar as superfícies equipotenciais devido a hipótese utilizada para o número de Peclet ($Pe \gg 2$), além de mostrar a distribuição radial da temperatura no disco de freio em quatro instantes diferentes. Utilizou-se o *software* aberto *Paraview* para a plotagem do resultado usando o filtro de revolução de uma seção do programa.

Percebe-se que a região de contato entre a pastilha e o disco é região em que as temperaturas são mais acentuadas devido ao fluxo de calor q imposto pelo atrito entre os componentes durante a frenagem. A temperatura nas outras regiões aumenta com o tempo de forma mais lenta, fenômeno explicado pela transferência de calor por condução.

Além disso, se observa que a temperatura na região de contato entre pastilha-disco é mais elevada para posições mais distantes do centro do disco, ou seja, para maiores coordenadas radiais. Isso ocorre devido a hipótese adotada de distribuição de pressão uniforme no disco de freio. Nesse modelo, como visto no Capítulo 3, a pressão é função da coordenada radial e do tempo.

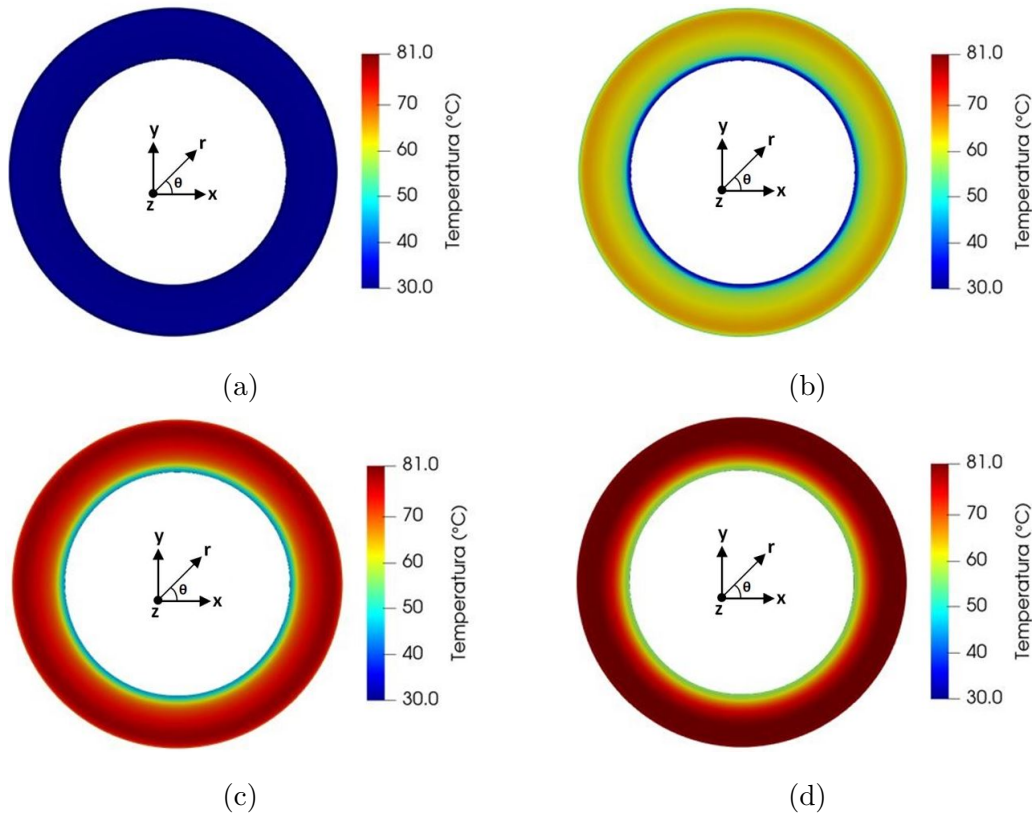


Figura 6.4: Visualização da simulação numérica para o disco de freio no plano $z = 2mm$, utilizando o *software Paraview*, nos instantes (a) $t = 0s$, (b) $t = 0.75s$, (c) $t = 1.5s$ e (d) $t = 2.65s$ considerando $t_b = 2.65s$

As Figuras 6.5-6.8 mostram detalhadamente diferentes configurações de resultados para os valores de temperatura no disco de freio. A Figura 6.5 mostra os valores de temperatura na extremidade do disco durante o tempo de frenagem para diferentes coordenadas radiais. Percebe-se que na região de contato pastilha-disco ($r = 65mm$ e $r = 75mm$), ocorre um elevado aumento de temperatura devido o fluxo de calor q gerado na região e, posteriormente, decresce devido aos efeitos convectivos governados pela *Lei de Resfriamento de Newton*.

Na região acima da área de geração de calor ($r = 83mm$), o aumento da temperatura nos instantes iniciais é mais suave, pois o calor é transferido por condução para essa região que constantemente troca calor com o fluido ambiente por convecção. Enquanto que na região abaixo da área de geração de calor ($r = 56.5mm$), o aumento da temperatura é gradual e menos acentuada, pois os efeitos convectivos são menores e o calor é transferido por condução.

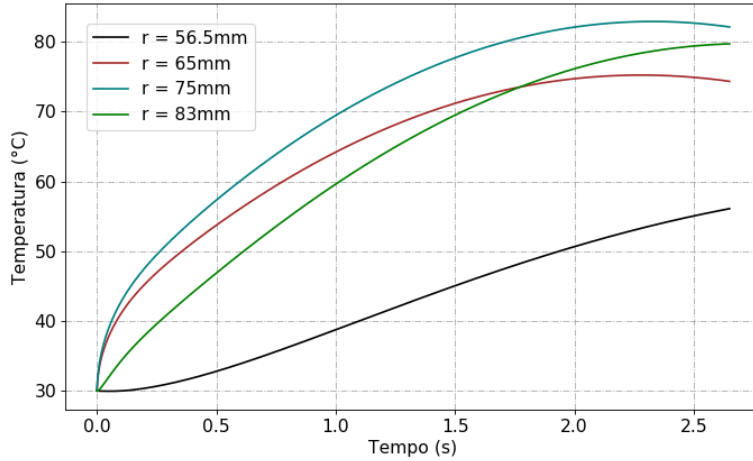


Figura 6.5: Temperatura no disco em diferentes coordenadas radiais em $z = \delta_d$

A Figura 6.6 ilustra os valores de temperatura no meio ($z = 0$) e na extremidade do disco ($z = 2mm$) após a frenagem para diferentes coordenadas radiais. Observa-se valores de temperatura mais elevados na região de contato entre os componentes ($58.5mm \leq r \leq 82.5mm$) conforme a coordenada radial aumenta. Na região próxima ao raio externo do disco ($r = 83mm$) há um decréscimo de temperatura pouco acentuado devido os efeitos convectivos. Ainda, podemos notar que a temperatura na extremidade do disco ($z = 2mm$) é levemente mais acentuada que em seu centro, já que o disco possui uma espessura pequena, logo a transferência de calor por condução ocorre rapidamente.

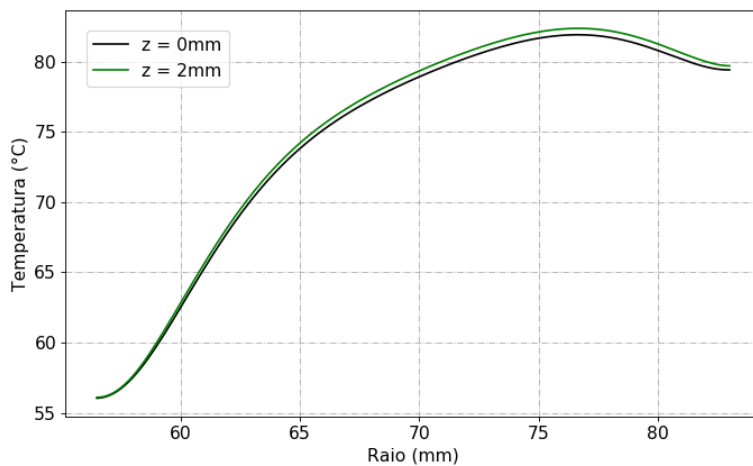


Figura 6.6: Temperatura no disco em diferentes coordenadas axiais em função da coordenada radial no tempo $t = t_b$

A figura abaixo mostra os valores de temperatura no meio ($z = 0$) e na extre-

midade do disco ($z = 2mm$) durante o tempo de frenagem na coordenada radial $r = 83mm$. Os valores de temperatura durante a frenagem tendem a ser maiores na extremidade do disco em relação ao seu centro, no entanto a temperatura das coordenadas tendem a convergir para um valor, pois a geração de calor e a transferência de calor por convecção diminuem com o tempo e os efeitos condutivos preponderam de forma a atingir o equilíbrio.

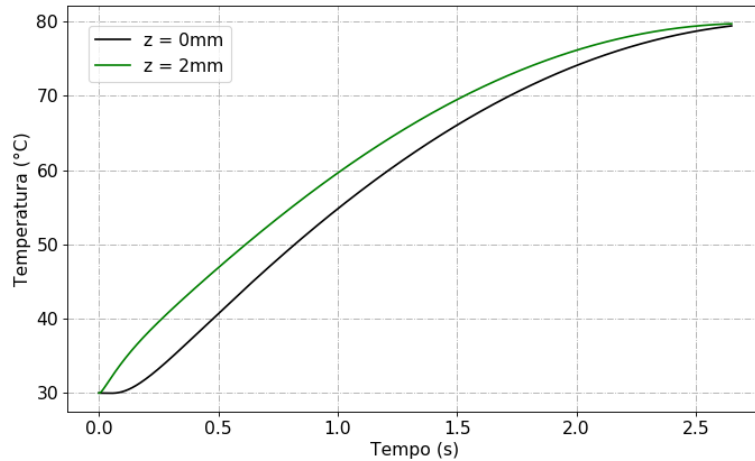


Figura 6.7: Temperatura no disco em diferentes coordenadas axiais em $r = 83mm$

Por fim, a Figura 6.8 ilustra os valores de temperatura no disco em diferentes coordenadas radiais ao longo de sua espessura após a frenagem.

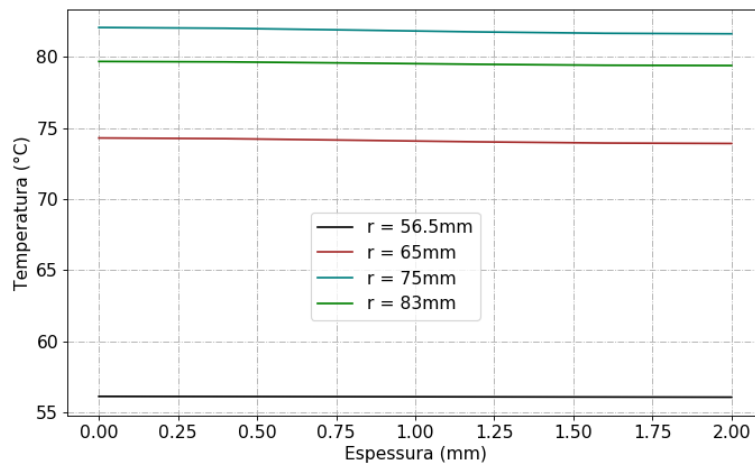


Figura 6.8: Temperatura no disco em diferentes coordenadas radiais ao longo de sua espessura no tempo $t = t_b$

Observa-se, novamente, que a variação da temperatura ao longo da espessura no

instante $t = t_b$ é pequena, mostrando um equilíbrio descrito no parágrafo anterior. Além disso, percebe-se valores de temperatura mais elevados na região em que há geração de calor q , como apresentado anteriormente.

As figuras abaixo mostram o avanço da temperatura na coordenada axial z em quatro instantes de tempo com objetivo de ilustrar o equilíbrio atingido apresentado nas figuras anteriores.

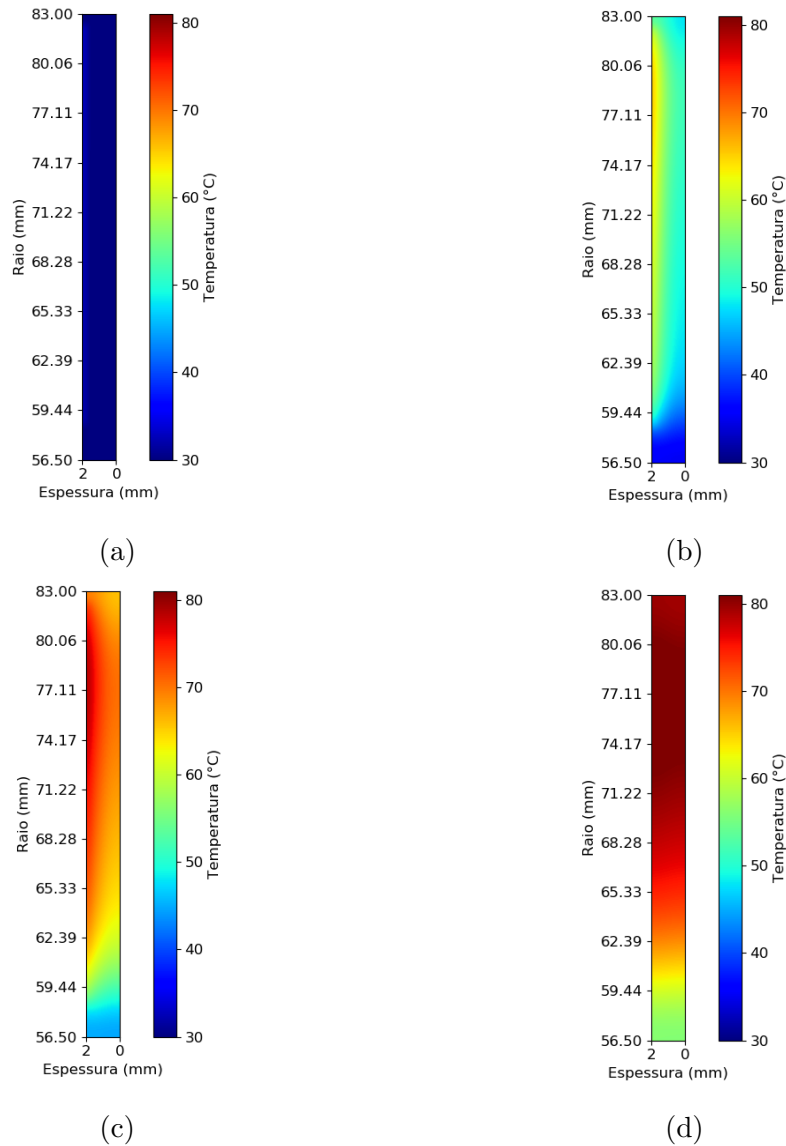


Figura 6.9: Avanço da temperatura na coordenada axial para frenagem de emergência nos instantes (a) $t = 0s$, (b) $t = 0.75s$, (c) $t = 1.5s$ e (d) $t = 2.65s$ considerando $t_b = 2.65s$

6.2 Múltiplas Frenagens

Para o modelo de múltiplas frenagens, utilizou-se os mesmo parâmetros geométricos e de projeto apresentados nas Tabelas 6.1 e 6.4 e as mesmas propriedades dos materiais do mecanismo disco-pastilha e do fluido (ar). Contudo, adicionou-se os seguinte parâmetros responsáveis pela etapa de resfriamento do disco (Tabela 6.6.):

Velocidade máxima do veículo (v_{max})	15	m/s
Tempo de aceleração (t_a)	4.3	s
Número de frenagens (n_b)	100	und.

Tabela 6.6: Parâmetros adicionais de projeto para a etapa de resfriamento do disco

Foram realizadas cinco simulações variando o tempo de resfriamento Δt de 10s à 50s. O algoritmo ocorre da seguinte forma:

- Frenagem de v_0 até o repouso durante o tempo t_b ;
- Aceleração do veículo até a velocidade máxima v_{max} durante o tempo de aceleração t_a ;
- Velocidade máxima constante durante o instante de tempo $\Delta t - t_a$;
- Frenagem de v_{max} até o repouso durante o tempo t_b , completando o ciclo n_b vezes.

Utilizou-se os mesmos parâmetros numéricos utilizados na simulação para frenagem de emergência, com a exceção do espaço de tempo dt que foi alterado para $dt = 0.01s$ para reduzir o tempo de processamento dos dados.

A Figura 6.10 mostra os valores de temperatura máxima no disco de freio para diferentes valores de Δt em função do número de frenagens. Observa-se dois comportamentos:

1. A temperatura máxima decresce com o aumento do tempo de resfriamento Δt , pois os efeitos da transferência de calor por convecção predominam, transferindo calor para o fluido ambiente [1, 12].

2. A temperatura máxima no disco possui uma tendência à convergir e se estabilizar em uma temperatura após determinado número de frenagens. Isso acontece pois o incremento de temperatura em cada frenagem diminui a ponto de ser desprezível após n_p frenagens [1, 12].

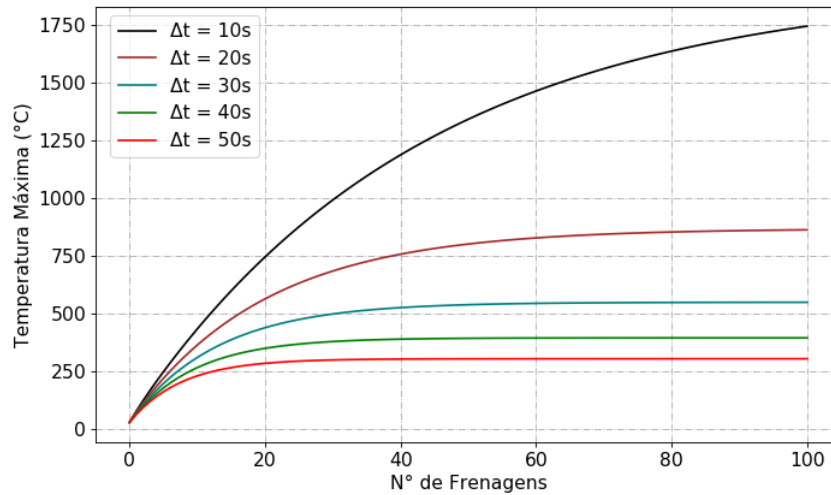


Figura 6.10: Temperatura máxima no disco de freio em função do número de frenagens n_p para diferentes valores de Δt

Para melhor compreensão do comportamento da temperatura no disco, avaliamos a temperatura no disco de freio durante o tempo de simulação em $r = 75mm$ e $z = 2mm$ e para $\Delta t = 20s$, conforme ilustrado na Figura 6.11.

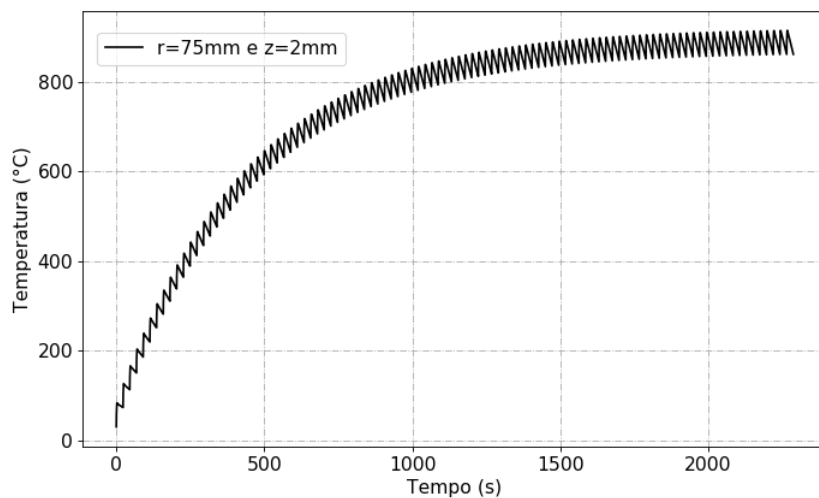


Figura 6.11: Temperatura no disco de freio em função do tempo de simulação para $\Delta t = 20s$

Verifica-se na figura anterior oscilações na temperatura decorrente do constante aquecimento e resfriamento do disco durante as frenagens. Além disso, nota-se que a temperatura converge para um valor máximo, como apresentado também na Figura 6.10.

Capítulo 7

Conclusão

Neste trabalho foi apresentado a equação de condução do calor em coordenadas cilíndricas utilizado em um modelo axissimétrico e com simetria axial numa abordagem do Método dos Elementos Finitos onde o esquema Taylor-Galerkin foi aplicado às equações de governo. Como a formulação não possuía acoplamento entre as variáveis, podemos utilizar o elemento triangular linear sem restrição possibilitando assim uma facilidade na implementação do código numérico além das variáveis envolvidas serem escalares e não vetoriais.

O código numérico apresentou resultados satisfatórios comparados às soluções analíticas do modelo clássico de um cilindro curto sujeito à efeitos convectivos, utilizado nesse trabalho como modelo de validação da estrutura computacional. As simulações realizadas mostraram que os erros em relação às soluções analíticas ficaram abaixo de 1% mesmo com a aproximação pelo raio médio na formulação numérica.

O objetivo desse trabalho foi desenvolver uma ferramenta numérica capaz de resolver o problema de transferência de calor em um sistema de freio à disco para dimensionamento térmico preliminar do disco de freio adotando uma geometria simples. Para isso foram realizadas simulações envolvendo situações de frenagens de emergência e múltiplas frenagens adotando parâmetros de projeto fornecidos pela Equipe Minerva Baja.

As simulações envolvendo a transferência de calor na interação pastilha-disco para as situações de frenagem de emergência e múltiplas frenagens se mostraram coerentes com o modelo físico proposto, respeitando as leis físicas e matemática

que caracterizam os fenômenos de transferência de calor em sólidos. Os valores da variável de interesse (temperatura) para os dados de projeto fornecidos se mantiveram dentro da ordem de grandeza esperada. Além disto, a discretização pelo MEF mostrou-se muito precisa para as simulações aqui apresentadas. É importante notar que o código foi totalmente desenvolvido em linguagem Python e é integralmente apresentado no apêndice deste documento.

O modelo e o algoritmo se encontram em constante desenvolvimento, portanto para trabalhos futuros destacamos:

- Utilização de uma geometria mais complexa, ou seja, a geometria real do disco de freio no modelo numérico;
- Utilização do modelo de desgaste uniforme para a pressão;
- Verificação do modelo para diferentes tipos de elementos e polinômios interpoladores;
- Criação de uma interface para o modelo numérico e, conseqüentemente, um *software*.

Referências Bibliográficas

- [1] LIMPERT, R., *Brake Design and Safety*. 3rd ed. SAE Internacional: Warrendale, PA - USA, 2011.
- [2] LIMPERT, R., *Brake Design and Safety*. 2nd ed. SAE Internacional: Warrendale, PA - USA, 1999.
- [3] LIMPERT, R., *Engineering Design Handbook: Analysis and design of automotive brake systems*. Darcom Pamphlet: USA, 1976.
- [4] TALATI, F., JALALIFAR, S., “Analysis of heat conduction in a disk brake system”, *Springer-Verlag*, v. 45, pp. 1047—1059, 2009.
- [5] ÖZISIK, M. N., *Heat Transfer, a Basic Approach*. 1st ed. McGraw-Hill Book Company: New York, EUA, 1985.
- [6] KUCIEJ, M., GRZES, P., “The Comparable Analysis of Temperature Distributions Assessment in Disc Brake Obtained Using Analytical Method and FE Model”, *Kones Powertrain and Transport*, v. 18, n. 2, pp. 1–16, 2011.
- [7] TALER, J., OCLON, P., *Finite Element Method in Steady-State and Transient Heat Conduction*. 1st ed. Spring Science: Dordrecht, NL, 2014.
- [8] INCROPERA, F. P., BERGMAN, T. L., LAVINE, A. S., et al., *Fundamentals of Heat and Mass Transfer*. 7th ed. John Wiley and Sons Ltd.: New Jersey, EUA, 2011.
- [9] LOGAN, D. L., *A First Course in the Finite Element Method*. 4th ed. Thomson Canada Ltd.: Toronto, Ontário - Canada, 2007.

- [10] ANJOS, G., *Computação Científica para Engenheiros*. PEM/COP-PE/UFRJ: Rio de Janeiro, RJ, Brasil, 2019.
- [11] NICOLAZZI, R., ROSA, E., LEAL, L., *Uma introdução à modelagem quase-estática de automóveis*. Universidade Federal de Santa Catarina (UFSC): Santa Catarina, ES, 2012.
- [12] ADAMOWICZ, A., GRZES, P., “Influence of Convective Cooling on a Disc Brake Temperature Distribution During Repetitive Braking”, *Applied Thermal Engineering*, v. 5, n. 2, pp. 2177–2185, 2011.
- [13] ADAMOWICZ, A., “Axisymmetric FE Model to analysis of thermal stresses in a brake disk”, *Journal of theoretical and applied mechanics*, v. 53, n. 2, pp. 357–370, 2014.
- [14] MAZID, H., JALALIFAR, S., “Mathematical modeling of heat conduction in a disk brake system during braking”, *Asian Journal of Applied Sciences*, v. 4, n. 2, pp. 119–136, 2011.
- [15] GRZES, P., “Partition of heat in 2d finite element model of a disc brake”, *Acta Mechanica et Automatica*, v. 4, n. 2, 2011.
- [16] GRZES, P., “Numerical analysis of temperature field in a disc brake at different cover angle of the pad”, *Acta Mechanica et Automatica*, v. 8, n. 4, 2014.
- [17] YEVTUSHENKO, A., GRZES, P., “The FEM-Modeling of the Frictional Heating Phenomenon in the Pad/Disc Tribosystem (A Review)”, *International Journal of Computation and Methodology*, v. 58, n. 3, pp. 207–226, 2010.
- [18] LEWIS, R., NITHIARASU, P., SEETHARAMU, K., *Fundamentals of the Finite Element Method for Heat and Fluid Flow*. 1st ed. John Wiley and Sons Ltd.: New Jersey, EUA, 2004.
- [19] PANTAKAR, S., *Numerical Heat Transfer and Fluid Flow*. 1st ed. McGraw-Hill Book Company: New York, EUA, 1980.

- [20] JOHNSON, C., *Numerical solution of partial differential equations by the finite element method*. 1st ed. CAMBRIDGE UNIVERSITY PRESS: Lund - Sweden, 1987.
- [21] REDDY, J., *An Introduction to the Finite Element Method*. 3rd ed. McGraw-Hill Book Company: Texas A&M University, EUA, 2006.
- [22] JAYAS, D., LALIBERT, G., ALAGUSUNDARAM, K., “A Finite Element Program for Teaching Transiente Axisymmetric Field Problems*”, *International Journal of Applied Engineering*, v. 7, n. 4, pp. 289–293, 1991.
- [23] RODRIGUES, J., *Cálculo de Transientes Térmicos Bidimensionais pelo Método de Elementos Finitos*, M.Sc.. Universidade de Santa Catarina: Florianópolis, Brasil, 1980.

Apêndice A

Código Fonte - Cilindro Curto

A.1 Solução Analítica

```
#Bibliotecas utilizadas no código
import numpy as np
import math

#Análise de um cilindro curto considerando:

#Bi = 1 e Fo = 0.4, 0.8 e 1.2

#Para Bi = 1, temos as constantes:

#A1 = 1.2071 e lambda1 = 1.2558

#DADOS DE ENTRADA

# 1. Geométricos
r = 0 #raio interno do cilindro em m
R = 1 #raio externo do cilindro em m
z = 1 #espessura do cilindro em m
```

2. Material do Cilindro

$k = 1$ *#condutividade térmica do material do
#cilindro em W/m C*

$c = 1$ *#calor específico do material do cilindro em J/kg C*

$\rho = 1$ *#massa específica do material do cilindro em kg/m³*

$\alpha = k/(\rho*c)$ *#difusividade térmica do material do
#cilindro em m²/s*

3. Interação Fluido - Cilindro

$T_a = 0$ *#temperatura ambiente externo em C*

$T_i = 1000$ *#temperatura inicial do cilindro em C*

$h = 1$ *#coeficiente de convecção em W/m² C*

$H = h/k$ *#razão dos coeficiente de convecção e condução em m⁻¹*

4. Parâmetros adimensionais

$A_1 = 1.2071$ *#constante 1*

$\lambda_1 = 1.2558$ *#constante 2*

$t = 0.4$ *#tempo em segundos*

$Bi = h*R/k$ *#número de Biot*

$Fo = \alpha*t/(R**2)$ *#número de Fourier*

$n = 29$ *#número de divisões do Raio*

#SOLUÇÃO ANALÍTICA APROXIMADA

$Raio = np.zeros(n+1)$ *#lista de raios em cada ponto*

$Raio[0] = 0$ *#origem do cilindro*

for i **in range** (1, n+1):

$Raio[i] = i*(R/n)$

#insere os raios na lista

$T = np.zeros(n+1)$ *#lista de temperaturas em cada ponto*

$T[0] = (T_i - T_a)*A_1*np.exp(-(\lambda_1**2)*Fo) + T_a$

```

#insere as temperaturas na lista
for i in range (1,n+1):
    J0 = 0 #define a função de Bessel
    for k in range (0,1000): #define a somatória
        J0 = J0 + (((-1)**k)
        /(math.factorial(k)**2))*
        ((lambda1*(i/(2*n)))**(2*k)) #Função de Bessel
    T[i] = (Ti-Ta)*A1*np.exp(-(lambda1**2)*Fo)*J0 + Ta

```

A.2 Solução Numérica

```

#Bibliotecas utilizadas no código
import numpy as np
import matplotlib.pyplot as plt

#DADOS DE ENTRADA

# 1. Geométricos
r = 0 #raio interno do cilindro em m
R = 1 #raio externo do cilindro em m
z = 1 #espessura do cilindro em m

# 2. Material do Cilindro
k = 1 #condutividade térmica do material do cilindro
    #em W/m C
c = 1 #calor específico do material do cilindro em J/kg C
rho = 1 #massa específica do material do cilindro em kg/m3
alpha = k/(rho*c) #difusividade térmica do material
    #do cilindro em m2/s

# 3. Interação Fluido - Cilindro
Ta = 0 #temperatura ambiente externo em C
Ti = 1000 #temperatura inicial do cilindro em C

```

```
h = 1 #coeficiente de convecção em W/m2 C
H = h/k #razão dos coeficiente de convecção e condução em m1
```

#PARÂMETROS DA SIMULAÇÃO

```
nz = 25 #número de nós na direção z
nr = 25 #número de nós na direção r
```

```
npoints = nr*nz #número de nós da malha
ne = 2*(nr-1)*(nz-1) #número de elementos
Lr = (R-r) #comprimento da seção na direção r em m
      #de (r<Raio<R)
Lz = z #comprimento da seção na direção z em m
```

#PRODUÇÃO DOS VETORES R E Z

```
zd = np.linspace(0,Lz,nz) #define a posição dos nós em z
rd = np.linspace(r,R,nr) #define a posição dos nós em r
      #(r<Raio<R)
```

```
Z1,R1 = np.meshgrid(zd,rd) #formação da malha
```

```
Z1 = Z1.reshape( (nz*nr) ) #reorganiza os pontos para
      #plotagem
```

```
R1 = R1.reshape( (nz*nr) ) #reorganiza os pontos para
      #plotagem
```

```
dr = (Lr)/(nr-1) #espaçamento da malha em r em m
dz = (Lz)/(nz-1) #espaçamento da malha em z em m
```

#CRIAÇÃO DOS VETORES RAI0 (R) E ESPESSURA (Z)

```
Raio = np.zeros((npoints), dtype='float')
```

```
a = 0 #incremento
```

```
for i in range(0, len (Raio)):
```

```
    Raio[i] = R1[a]
```

```
    a+=1
```

```
Z = np.zeros((npoints))
```

```
a = 0 #incremento
```

```
for i in range(0, len (Z1)):
```

```
    Z[i] = Z1[a]
```

```
    a+=1
```

#CRIAÇÃO DA IEN PARA CÁLCULO

```
IEN=np.zeros((ne,3), dtype='float') #criação da matriz IEN
```

```
element=0 #criação de um incremento
```

```
for j in range(0, nr-1):
```

```
    for i in range(0, nz-1):
```

```
        IEN[element]=[nz*j+i+1, nz*j+i, nz*(j+1)+i]
```

```
        IEN[element+1]=[nz*(j+1)+i, nz*(j+1)+i+1, nz*j+i+1]
```

```
        element+=2
```

#PLOT DA MALHA

```
plt.figure(figsize=(10, 6), dpi=80)
```

```
plt.scatter(Z1, R1, marker='o', s=50, color='red')
```

#plotagem dos nós

```
ax1 = plt.triplot(Z1, R1, IEN, color='black')
```

#plotagem da malha

```

plt.show()

#CÁLCULO DA ÁREA DOS ELEMENTOS

matriz = [[1,0,0],[1,dz,0],[1,0,dr]] #matriz das coordenadas
                                         #do elemento triangular
                                         #da malha

A = 0.5*np.linalg.det(matriz)
    #área do triângulo em m2 da malha

#CRIAÇÃO DO VETOR Rmédio

Rm = np.zeros((ne,1), dtype='float')
    #criação do vetor do raio médio da malha
elemento = 0

for j in range(0,nr-1):
    for i in range(0,nz-1):
        Rm[elemento] = (3*r+(3*j+1)*dr)/3
        Rm[elemento+1] = (3*r+(3*j+2)*dr)/3
        elemento+=2

#CRIAÇÃO DO VETOR B (gradiente)

Bi = np.dot((0.5/A),[[-dr,dr,0],[0,dz,-dz]])
Bit = np.transpose(Bi)

#MONTAGEM DA MATRIZ DE RIGIDEZ CONDUTIVA DO ELEMENTOS

kc = 2*np.pi*A*np.dot(Bit,Bi) #matriz rigidez condutiva dos
                                         #elementos da malha

Kc = np.zeros((npoints,npoints), dtype='float')

```

```

#cria a matriz rigidez condutiva da malha

for elem in range(0,ne):
    for ilocal in range(0,3):
        iglobal = int(IEN[elem,ilocal])
        for jlocal in range(0,3):
            jglobal = int(IEN[elem,jlocal])
            Kc[iglobal, jglobal] = Kc[iglobal, jglobal] +
            Rm[elem,0]*kc[ilocal, jlocal]

#MONTAGEM DA MATRIZ DE RIGIDEZ CONVECTIVA

kh = np.dot((2*np.pi*dz*H/12),
[[4*R,2*R,0],[2*R,4*R,0],[0,0,0]]) #matriz rigidez
#convectiva na borda superior da malha
Kh = np.zeros((npoints, npoints), dtype='float')
#cria a matriz rigidez convectiva global na
#borda superior da malha (ij)

nh = [] #cria os vetores dos elementos convectivos
#na borda superior
for i in range(ne-(2*nz-3),ne+1,2):
    nh.append(i)

for elem in nh:
    for ilocal in range(0,3):
        iglobal = int(IEN[elem,ilocal])
        for jlocal in range(0,3):
            jglobal = int(IEN[elem,jlocal])
            Kh[iglobal, jglobal] = Kh[iglobal, jglobal] +
            kh[ilocal, jlocal]

```

#MONTAGEM DA MATRIZ RIGIDEZ GLOBAL

$K = K_c + K_h$

#MONTAGEM DA MATRIZ DE MASSA (OU CAPACITÂNCIA) DOS ELEMENTOS

$m = np.dot((2*np.pi*A/(12*alpha)), [[2,1,1],[1,2,1],[1,1,2]])$
#matriz de massa do elemento da malha

$M = np.zeros((npoints, npoints), dtype='float')$
#cria a matriz de massa da malha 1

```
for elem in range(0,ne):  
    for ilocal in range(0,3):  
        iglobal = int(IEN[elem, ilocal])  
        for jlocal in range (0,3):  
            jglobal = int(IEN[elem, jlocal])  
            M[iglobal, jglobal] = M[iglobal, jglobal] +  
                Rm[elem,0]*m[ilocal, jlocal]
```

#MONTAGEM DO VETOR FORÇA - CONVECÇÃO

$fh = np.dot((2*np.pi*dz*H*Ta/6), [[3*R],[3*R],[0]])$
#cria o vetor força convectiva do elemento
#na malha na borda superior

$Fh = np.zeros((npoints,1))$ *#cria o vetor força convectiva*

```
for elem in nh:  
    for jlocal in range(0,3):  
        jglobal = int(IEN[elem, jlocal])
```

```
Fh[jglobal,0] = Fh[jglobal, 0] + fh[jlocal,0]
```

```
#MONTAGEM DO VETOR FORÇA INICIAL
```

```
Fi = np.zeros((npoints,1), dtype='float')  
for i in range(0,npoints):  
    Fi[i]+=Ti
```

```
#SOLUÇÃO DO SISTEMA LINEAR
```

```
#dt = np.min((dr1, dr2, dr3, dz)**2/(10*alpha))  
    #espaço de tempo em segundos  
dt = 0.001 #espaço de tempo em segundos  
theta = 0.5 #discretização temporal (0 = explicito ,  
    #0.5 = cranck-nicolson, 1 = implicito)  
T = Fi #temperatura inicial em C
```

```
#MONTAGEM DA EQUAÇÃO (ASSEMBLY)
```

```
A = M/dt + theta*K #matriz que compõe o tempo n+1  
B = M/dt - (1-theta)*K #matriz que compõe o tempo n
```

```
for i in range(0,401):  
    b = np.dot(B,T) + Fh #vetor solução  
    T = np.linalg.solve(A,b) #solução numérica
```

Apêndice B

Código Fonte - Dimensionamento do Disco de Freio

B.1 Frenagem de Emergência

```
#Bibliotecas utilizadas no código
import numpy as np
import matplotlib.pyplot as plt

#DADOS DE ENTRADA

# 1. Geométricos
rd = 56.5 #raio interno do disco (furo) em mm
rp = 58.5 #raio interno da pastilha em mm
Rp = 82.5 #raio externo da pastilha em mm
Rd = 83 #raio externo do disco em mm
d = 4 #espessura do disco em mm
p = 3 #espessura da pastilha e mm
ang = 60 #ângulo da área de contato entre pastilha
       #e disco (estacionário) em graus

# 2. Material do Disco
kd = 42 #condutividade térmica do material do disco em W/m C
```

cd = 469 *#calor específico do material do disco em J/kg C*
rhod = 7200 *#massa específica do material do disco em kg/m³*
alpha = kd/(rhod*cd) *#difusividade térmica do material do*
#disco em m²/s

3. Material da Pastilha

kp = 12 *#condutividade térmica do material da pastilha*
#em W/m C
cp = 900 *#calor específico do material da pastilha em J/kg C*
rhop = 2500 *#massa específica do material da pastilha*
#em kg/m³

4. Informações de Projeto

Rr = 0.267 *#raio efetivo da roda em m*
tb = 2.65 *#tempo de parada em s*
Pm = 3.64*(10**6) *#pressão máxima aplicada pela pastilha*
#em Pa (N/m²)

5. Interação Pastilha - Disco

mi = 0.40 *#coeficiente de atrito pastilha-disco*
gamma = 1/(1+np.sqrt(rhop*cp*kp/(rhod*cd*kd)))
#coeficiente de partição de calor
phi = np.pi*ang/180
#ângulo da área de contato entre pastilha e disco em rad
Sp = np.pi*(Rp**2 - rp**2)
#área da superfície de contato pastilha-disco em mm²

6. Interação Fluido - Disco E Fluxo de calor inicial

Tf= 25 *#temperatura ambiente do ar em C*
Ti = 30 *#temperatura inicial do disco em C*
kar = 0.0243 *#condutividade Térmica do ar em W/m C*
vis = 10**(-5) *#viscosidade cinemática do ar em m²/s*

#PARÂMETROS DA SIMULAÇÃO

```
nz = 6 #número de nós na direção z
nr2 = 49*2 #número de nós na direção r de (rp<r<Rp)
nr1 = int((((rp-rd)/(Rp-rp))*(nr2-1))+1)
      #número de nós na direção r de (rd<r<rp)
nr3 = int((((Rd-Rp)/(Rp-rp))*(nr2-1))+1)
      #número de nós na direção r de (Rp<r<Rd)
nr = nr1+nr2+nr3-2 #número total de nós na direção r

npoints1 = nr1*nz #número de pontos da malha 1
nel = 2*(nr1-1)*(nz-1) #número de elementos da malha 1
Lr1 = (rp-rd) #comprimento da seção na direção r
      #em mm de (rd<r<rp)

npoints2 = nr2*nz #número de pontos da malha 2
ne2 = 2*(nr2-1)*(nz-1) #número de elementos da malha 2
Lr2 = (Rp-rp) #comprimento da seção na direção r
      #em mm de (Rp<r<rp)

npoints3 = nr3*nz #número de pontos da malha 3
ne3 = 2*(nr3-1)*(nz-1) #número de elementos da malha 3
Lr3 = (Rd-Rp) #comprimento da seção na direção r
      #em mm de (Rp<r<Rd)

Lz = d/2 #comprimento da seção na direção z em mm

npoints = (npoints1 + npoints2 + npoints3)-2*nz
      #número total de nós da malha geral
ne = nel + ne2 + ne3
      #número total de elementos da malha geral
```

#PRODUÇÃO DOS VETORES X E Y

```
zd = np.linspace(0,Lz,nz) #define a posição dos nós em z
r1d = np.linspace(rd,rp,nr1) #define a posição dos nós
                                #em (rd<r<rp)
r2d = np.linspace(rp,Rp,nr2) #define a posição dos nós
                                #em (rp<r<Rp)
r3d = np.linspace(Rp,Rd,nr3) #define a posição dos nós
                                #em (Rp<r<Rd)

Z1,R1 = np.meshgrid(zd,r1d) #formação da malha 1
Z2,R2 = np.meshgrid(zd,r2d) #formação da malha 2
Z3,R3 = np.meshgrid(zd,r3d) #formação da malha 3

Z1 = Z1.reshape( (nz*nr1) ) #reorganiza os pontos
                                #para plotagem
Z2 = Z2.reshape( (nz*nr2) ) #reorganiza os pontos
                                #para plotagem
Z3 = Z3.reshape( (nz*nr3) ) #reorganiza os pontos
                                #para plotagem
R1 = R1.reshape( (nz*nr1) ) #reorganiza os pontos
                                #para plotagem
R2 = R2.reshape( (nz*nr2) ) #reorganiza os pontos
                                #para plotagem
R3 = R3.reshape( (nz*nr3) ) #reorganiza os pontos
                                #para plotagem

dr1 = (Lr1/1000)/(nr1-1) #espaçamento da malha em r
dr2 = (Lr2/1000)/(nr2-1) #espaçamento da malha em r
dr3 = (Lr3/1000)/(nr3-1) #espaçamento da malha em r
dz = (Lz/1000)/(nz-1) #espaçamento da malha em z
```

#CRIAÇÃO DOS VETORES RAIOS (R) E ESPESSURA (Z)

```
R = np.zeros((npoints), dtype='float')
```

```
a = 0 #incremento
```

```
for i in range(0, len(R1)):
```

```
    R[i] = R1[a]
```

```
    a+=1
```

```
a = nz #incremento
```

```
for i in range(len(R1), len(R2)-nz + len(R1)):
```

```
    R[i] = R2[a]
```

```
    a+=1
```

```
a = nz #incremento
```

```
for i in range(len(R2)-nz + len(R1), len(R2)-2*nz +  
               len(R1) + len(R3)):
```

```
    R[i] = R3[a]
```

```
    a+=1
```

```
Z = np.zeros((npoints))
```

```
a = 0 #incremento
```

```
for i in range(0, len(Z1)):
```

```
    Z[i] = Z1[a]
```

```
    a+=1
```

```
a = nz #incremento
```

```
for i in range(len(Z1), len(Z2)-nz + len(Z1)):
```

```
    Z[i] = Z2[a]
```

```
    a+=1
```

```

a = 0 #incremento
for i in range (len(Z2)-nz + len (Z1), len(Z2)-2*nz +
                len (Z1) + len (Z3)):
    Z[i] = Z3[a]
    a+=1

```

#CRIAÇÃO DA IEN PARA CÁLCULO

```

IEN1=np.zeros((ne,3), dtype='float')
    #criação da matriz IEN da malha 1
element=0 #criação de um incremento
for j in range(0,nr1-1):
    for i in range(0,nz-1):
        IEN1[element]=[nz*j+i+1,nz*j+i ,nz*(j+1)+i]
        IEN1[element+1]=[nz*(j+1)+i ,nz*(j+1)+i+1,nz*j+i+1]
        element+=2

```

```

IEN2=np.zeros((ne,3)) #criação da matriz IEN da malha 2
for j in range(nr1-1,nr1+nr2-2):
    for i in range(0,nz-1):
        IEN2[element]=[nz*j+i+1,nz*j+i ,nz*(j+1)+i]
        IEN2[element+1]=[nz*(j+1)+i ,nz*(j+1)+i+1,nz*j+i+1]
        element+=2

```

```

IEN3=np.zeros((ne,3)) #criação da matriz IEN da malha 3
for j in range(nr2+nr1-2,nr2+nr1+nr3-3):
    for i in range(0,nz-1):
        IEN3[element]=[nz*j+i+1,nz*j+i ,nz*(j+1)+i]
        IEN3[element+1]=[nz*(j+1)+i ,nz*(j+1)+i+1,nz*j+i+1]
        element+=2

```

```
IEN = IEN1 + IEN2 + IEN3
```

```
#CRIAÇÃO DA IEN PARA PLOTAGEM
```

```
IEN1p=np.zeros((ne,3),dtype='float') #criação da matriz IEN  
element=0 #criação de um incremento
```

```
for j in range(0,nr1-1):
```

```
    for i in range(0,nz-1):
```

```
        IEN1p[element]=[nz*j+i+1,nz*j+i ,nz*(j+1)+i]
```

```
        IEN1p[element+1]=[nz*(j+1)+i ,nz*(j+1)+i+1,nz*j+i+1]
```

```
        element+=2
```

```
IEN2p=np.zeros((ne,3)) #criação da matriz IEN
```

```
for j in range(0,nr2-1):
```

```
    for i in range(0,nz-1):
```

```
        IEN2p[element]=[nz*j+i+1,nz*j+i ,nz*(j+1)+i]
```

```
        IEN2p[element+1]=[nz*(j+1)+i ,nz*(j+1)+i+1,nz*j+i+1]
```

```
        element+=2
```

```
IEN3p=np.zeros((ne,3)) #criação da matriz IEN
```

```
for j in range(0,nr3-1):
```

```
    for i in range(0,nz-1):
```

```
        IEN3p[element]=[nz*j+i+1,nz*j+i ,nz*(j+1)+i]
```

```
        IEN3p[element+1]=[nz*(j+1)+i ,nz*(j+1)+i+1,nz*j+i+1]
```

```
        element+=2
```

```
#PLOT DA MALHA
```

```
plt.figure(figsize=(10, 6), dpi=80)
```

```
plt.scatter(Z1,R1,marker='o',s=50,color='red')
```

```
    #plotagem dos nós
```

```
plt.scatter(Z2,R2,marker='o',s=50,color='red')
```

```

    #plotagem dos nós
plt.scatter(Z3,R3,marker='o',s=50,color='red')
    #plotagem dos nós
ax1 = plt.triplot(Z1,R1,IEN1p, color='black')
    #plotagem da malha 1
ax2 = plt.triplot(Z2,R2,IEN2p, color='black')
    #plotagem da malha 2
ax3 = plt.triplot(Z3,R3,IEN3p, color='black')
    #plotagem da malha 3
plt.show()

```

#CÁLCULO DAS ÁREAS DOS ELEMENTOS

```

matriz1 = [[1,0,0],[1,dz,0],[1,0,dr1]] #matriz das
    #coordenadas do elemento triangular da malha 1
matriz2 = [[1,0,0],[1,dz,0],[1,0,dr2]] #matriz das
    #coordenadas do elemento triangular da malha 2
matriz3 = [[1,0,0],[1,dz,0],[1,0,dr3]] #matriz das
    #coordenadas do elemento triangular da malha 3
A1 = 0.5*np.linalg.det(matriz1) #área do triângulo
    #em m2 da malha 1
A2 = 0.5*np.linalg.det(matriz2) #área do triângulo
    #em m2 da malha 2
A3 = 0.5*np.linalg.det(matriz3) #área do triângulo
    #em m2 da malha 3

```

#CRIAÇÃO DO VETOR Rmédio

```

Rm1 = np.zeros((ne,1),dtype='float') #criação do vetor
    #do raio médio da malha 1
Rm2 = np.zeros((ne,1),dtype='float') #criação do vetor
    #do raio médio da malha 2

```

```
Rm3 = np.zeros((ne,1), dtype='float') #criação do vetor
      #do raio médio da malha 3
elemento = 0
```

```
for j in range (0, nr1-1):
    for i in range (0, nz-1):
        Rm1 [elemento] = (3*0.001*rd+(3*j+1)*dr1)/3
        Rm1 [elemento+1] = (3*0.001*rd+(3*j+2)*dr1)/3
        elemento+=2
```

```
for j in range (0, nr2-1):
    for i in range (0, nz-1):
        Rm2 [elemento] = (3*0.001*rd+3*(nr1-1)*dr1+
        (3*j+1)*dr2)/3
        Rm2 [elemento+1] = (3*0.001*rd+3*(nr1-1)*dr1+
        (3*j+2)*dr2)/3
        elemento+=2
```

```
for j in range (0, nr3-1):
    for i in range (0, nz-1):
        Rm3 [elemento] = (3*0.001*rd+3*(nr1-1)*dr1
        +3*(nr2-1)*dr2+(3*j+1)*dr3)/3
        Rm3 [elemento+1] = (3*0.001*rd+3*(nr1-1)*dr1
        +3*(nr2-1)*dr2+(3*j+2)*dr3)/3
        elemento+=2
```

```
Rm = Rm1 + Rm2 + Rm3
```

```
#CRIAÇÃO DO VETOR B (gradiente)
```

```
Bi1 = np.dot((0.5/A1), [[-dr1, dr1, 0], [0, dz, -dz]])
Bit1 = np.transpose(Bi1)
```

```
Bi2 = np.dot((0.5/A2), [[-dr2, dr2, 0], [0, dz, -dz]])
```

```
Bit2 = np.transpose(Bi2)
```

```
Bi3 = np.dot((0.5/A3), [[-dr3, dr3, 0], [0, dz, -dz]])
```

```
Bit3 = np.transpose(Bi3)
```

```
#MONTAGEM DA MATRIZ DE RIGIDEZ CONDUTIVA DO ELEMENTOS
```

```
kc1 = 2*np.pi*A1*np.dot(Bit1, Bi1) #matriz rigidez  
#condutiva dos elementos da malha 1
```

```
kc2 = 2*np.pi*A2*np.dot(Bit2, Bi2) #matriz rigidez  
#condutiva dos elementos da malha 2
```

```
kc3 = 2*np.pi*A3*np.dot(Bit3, Bi3) #matriz rigidez  
#condutiva dos elementos da malha 3
```

```
K1 = np.zeros((npoints, npoints), dtype='float')  
#cria a matriz rigidez condutiva da malha 1
```

```
K2 = np.zeros((npoints, npoints), dtype='float')  
#cria a matriz rigidez condutiva da malha 2
```

```
K3 = np.zeros((npoints, npoints), dtype='float')  
#cria a matriz rigidez condutiva da malha 3
```

```
for elem in range(0, ne1):
```

```
    for ilocal in range(0, 3):
```

```
        iglobal = int(IEN1[elem, ilocal])
```

```
        for jlocal in range(0, 3):
```

```
            jglobal = int(IEN1[elem, jlocal])
```

```
            K1[iglobal, jglobal] = K1[iglobal, jglobal] +
```

```
                Rm[elem, 0]*kc1[ilocal, jlocal]
```

```
for elem in range(ne1, ne1+ne2):
```

```

for ilocal in range(0,3):
    iglobal = int(IEN2[elem,ilocal])
    for jlocal in range (0,3):
        jglobal = int(IEN2[elem,jlocal])
        K2[iglobal , jglobal] = K2[iglobal , jglobal] +
            Rm[elem,0]*kc2[ilocal , jlocal]

```

```

for elem in range(ne1+ne2,ne1+ne2+ne3):
    for ilocal in range(0,3):
        iglobal = int(IEN3[elem,ilocal])
        for jlocal in range (0,3):
            jglobal = int(IEN3[elem,jlocal])
            K3[iglobal , jglobal] = K3[iglobal , jglobal] +
                Rm[elem,0]*kc3[ilocal , jlocal]

```

$K_c = K_1 + K_2 + K_3$

#MONTAGEM DA MATRIZ DE MASSA (OU CAPACITÂNCIA) DOS ELEMENTOS

```

me1 = np.dot(2*np.pi*A1/(12*alpha),
             [[2,1,1],[1,2,1],[1,1,2]])
#matriz de massa do elemento da malha 1

```

```

me2 = np.dot(2*np.pi*A2/(12*alpha),
             [[2,1,1],[1,2,1],[1,1,2]])
#matriz de massa do elemento da malha 2

```

```

me3 = np.dot(2*np.pi*A3/(12*alpha),
             [[2,1,1],[1,2,1],[1,1,2]])
#matriz de massa do elemento da malha 3

```

```

M1 = np.zeros((npoints ,npoints),dtype='float')
#cria a matriz de massa da malha 1

```

```

M2 = np.zeros((npoints ,npoints),dtype='float')

```

```

    #cria a matriz de massa da malha 2
M3 = np.zeros((npoints ,npoints),dtype='float ')
    #cria a matriz de massa da malha 3

for elem in range(0,ne1):
    for ilocal in range(0,3):
        iglobal = int(IEN1[elem,ilocal])
        for jlocal in range (0,3):
            jglobal = int(IEN1[elem,jlocal])
            M1[iglobal , jglobal] = M1[iglobal , jglobal] +
            Rm[elem,0]*me1[ilocal , jlocal]

for elem in range(ne1,ne1+ne2):
    for ilocal in range(0,3):
        iglobal = int(IEN2[elem,ilocal])
        for jlocal in range (0,3):
            jglobal = int(IEN2[elem,jlocal])
            M2[iglobal , jglobal] = M2[iglobal , jglobal] +
            Rm[elem,0]*me2[ilocal , jlocal]

for elem in range(ne1+ne2,ne1+ne2+ne3):
    for ilocal in range(0,3):
        iglobal = int(IEN3[elem,ilocal])
        for jlocal in range (0,3):
            jglobal = int(IEN3[elem,jlocal])
            M3[iglobal , jglobal] = M3[iglobal , jglobal] +
            Rm[elem,0]*me3[ilocal , jlocal]

```

#MONTAGEM DA MATRIZ DE MASSA (OU CAPACITÂNCIA) TOTAL

$M = M1 + M2 + M3$

```
#MONTAGEM DO VETOR FORÇA INICIAL
```

```
Fi = np.zeros((npoints,1), dtype='float')
```

```
for i in range(0,npoints):
```

```
    Fi[i]+=Ti
```

```
#SOLUÇÃO DO SISTEMA LINEAR
```

```
dt = 0.001 #espaço de tempo em s
```

```
theta = 0.5 #discretização temporal(0 = explícito,
```

```
    #0.5 = cranck-nicolson, 1 = implícito)
```

```
T = Fi
```

```
#FRENAGEM DE EMERGÊNCIA
```

```
for i in range(0,int(1/dt)+1):
```

```
    tempo = i*(tb*dt)
```

```
    #MONTAGEM DA MATRIZ DE RIGIDEZ CONVECTIVA
```

```
    v = 15 #velocidade máxima do veículo em m/s
```

```
    omega = (v/Rr)*(1-(tempo/tb)) #velocidade do disco
```

```
        #de freio em rad/s
```

```
    Re1 = (omega/vis)*(2*((rp+rd)/2000))**2 #número de
```

```
    #Reynolds para convecção na borda esquerda
```

```
    #da malha 1 (jk)
```

```
    Re2 = (omega/vis)*(2*((Rd+Rp)/2000))**2 #número de
```

```
    #Reynolds para convecção na borda esquerda
```

```
    #da malha 3 (jk)
```

```
    Re3 = (omega/vis)*(2*(Rd/1000))**2 #número de
```

```
    #Reynolds para convecção na borda superior
```

```
    #da malha 3 (ij)
```

```

if Re1 <= 10**6: #definição do coeficiente de
                    #convecção em W/m2 C
    h1 = (0.7*kar/(2*((rp+rd)/2000)))*(Re1**0.55)
else:
    h1 = (0.4*kar/(2*((rp+rd)/2000)))*(Re1**0.8)

if Re2 <= 10**6: #definição do coeficiente de
                    #convecção em W/m2 C
    h2 = (0.7*kar/(2*((Rd+Rp)/2000)))*(Re2**0.55)
else:
    h2 = (0.4*kar/(2*((Rd+Rp)/2000)))*(Re2**0.8)

if Re3 <= 10**6: #definição do coeficiente de
                    #convecção em W/m2 C
    h3 = (0.7*kar/(2*(Rd/1000)))*(Re3**0.55)
else:
    h3 = (0.4*kar/(2*(Rd/1000)))*(Re3**0.8)

if omega == 0: #convecção natural em W/m2 C
    h1 = 5
    h2 = 5
    h3 = 5

H1 = h1/kd #razão utilizada para simplificação
#dos cálculos: m1 - borda esquerda da malha 1 (jk)
H2 = h2/kd #razão utilizada para simplificação
#dos cálculos: m1 - borda esquerda da malha 3 (jk)
H3 = h3/kd #razão utilizada para simplificação
#dos cálculos: m1 - borda superior da malha 3 (ij)

kh11 = np.dot((2*np.pi*dr1*H1/12),

```

```

        [[0,0,0],[0,4,2],[0,2,4]])
#matriz rigidez convectiva na borda esquerda
#da malha 1 (jk)
kh13 = np.dot((2*np.pi*dr3*H2/12),
        [[0,0,0],[0,4,2],[0,2,4]])
#matriz rigidez convectiva na borda esquerda
#da malha 3 (jk)
kh23 = np.dot((2*np.pi*dz*H3/12),
[[4*Rd/1000,2*Rd/1000,0],[2*Rd/1000,4*Rd/1000,0]
,[0,0,0]])
#matriz rigidez convectiva na borda superior
#da malha 3 (ij)

Kh11 = np.zeros((npoints,npoints),dtype='float')
#cria a matriz rigidez convectiva global
#na borda esquerda da malha 1(jk)
Kh13 = np.zeros((npoints,npoints),dtype='float')
#cria a matriz rigidez convectiva global
#na borda esquerda da malha 3 (jk)
Kh23 = np.zeros((npoints,npoints),dtype='float')
#cria a matriz rigidez convectiva global
#na borda superior da malha 3 (ij)

nh11 = [] #cria os vetores dos elementos convectivos
#na borda esquerda da malha 1
for i in range(0,ne1,2*(nz-1)):
    nh11.append(i)

nh13 = [] #cria os vetores dos elementos convectivos
#na borda esquerda da malha 3
for i in range((ne1+ne2),ne,2*(nz-1)):
    nh13.append(i)

```

```

nh23 = [] #cria os vetores dos elementos convectivos
         #na borda superior
for i in range (ne-(2*nz-3),ne+1,2):
    nh23.append(i)

for elem in nh11:
    for ilocal in range(0,3):
        iglobal = int(IEN[elem,ilocal])
        for jlocal in range (0,3):
            jglobal = int(IEN[elem,jlocal])
            Kh11[iglobal, jglobal] = Kh11[iglobal,
            jglobal] + Rm[elem,0]*kh11[ilocal, jlocal]

for elem in nh13:
    for ilocal in range(0,3):
        iglobal = int(IEN[elem,ilocal])
        for jlocal in range (0,3):
            jglobal = int(IEN[elem,jlocal])
            Kh13[iglobal, jglobal] = Kh13[iglobal,
            jglobal] + Rm[elem,0]*kh13[ilocal, jlocal]

for elem in nh23:
    for ilocal in range(0,3):
        iglobal = int(IEN[elem,ilocal])
        for jlocal in range (0,3):
            jglobal = int(IEN[elem,jlocal])
            Kh23[iglobal, jglobal] = Kh23[iglobal,
            jglobal] + kh23[ilocal, jlocal]

Kh = Kh11 + Kh13 + Kh23
     #matriz rigidez convectiva global

```

#MONTAGEM DA MATRIZ RIGIDEZ TOTAL

$$K = K_c + K_h$$

#MONTAGEM DO VETOR FORÇA - CONVECÇÃO

```
fh11 = np.dot((2*np.pi*dr1*H1*Tf)/6, [[0],[3],[3]])
```

```
#cria o vetor força convectiva do elemento
```

```
#na malha 1 na borda esquerda
```

```
fh13 = np.dot((2*np.pi*dr3*H2*Tf)/6, [[0],[3],[3]])
```

```
#cria o vetor força convectiva do elemento
```

```
#na malha 3 na borda esquerda
```

```
fh23 = np.dot((2*np.pi*dz*H3*Tf)/6,
```

```
[[3*Rd/1000],[3*Rd/1000],[0]])
```

```
#cria o vetor força convectiva do elemento
```

```
#na malha 3 na borda superior
```

```
fh = np.zeros((npoints,1))
```

```
#cria o vetor força convectiva
```

```
for elem in nh11:
```

```
    for jlocal in range(0,3):
```

```
        jglobal = int(IEN[elem,jlocal])
```

```
        fh[jglobal,0] = fh[jglobal, 0] +
```

```
        Rm[elem,0]*fh11[jlocal,0]
```

```
for elem in nh13:
```

```
    for jlocal in range(0,3):
```

```
        jglobal = int(IEN[elem,jlocal])
```

```
        fh[jglobal,0] = fh[jglobal, 0] +
```

```

Rm[elem,0]*fh13[jlocal,0]

for elem in nh23:
    for jlocal in range(0,3):
        jglobal = int(IEN[elem,jlocal])
        fh[jglobal,0] = fh[jglobal,0] +
            Rm[elem,0]*fh23[jlocal,0]

#MONTAGEM DO VETOR FORÇA - FLUXO DE CALOR INICIAL

qi = (1/kd)*(phi/(2*np.pi))*gamma*mi*Pm*omega
    #fluxo de calor inicial em W/m2 (t=0)

fq2 = np.dot((2*np.pi*dr2*qi)/6,[[0],[3],[3]])

nh12 = [] #cria os vetores dos elementos de
    #fluxo na borda esquerda da malha 2
for i in range(ne1,ne1+ne2,2*(nz-1)):
    nh12.append(i)

fq = np.zeros((npoints,1),dtype='float')
    #cria o vetor força por fluxo de calor

for elem in nh12:
    for jlocal in range(0,3):
        jglobal = int(IEN[elem,jlocal])
        fq[jglobal,0] = fq[jglobal,0] +
            (Rm[elem,0])*fq2[jlocal,0]*(R[jglobal]/1000)

#MONTAGEM DA EQUAÇÃO (ASSEMBLY)

A = M/dt + theta*K #matriz que compõe o tempo n+1

```

```
B = M/dt - (1-theta)*K #matriz que compõe o tempo n
```

```
b = np.dot(B,T) + fh - fq #vetor solução
```

```
T = np.linalg.solve(A,b) #solução numérica
```

B.2 Múltiplas Frenagens

O código fonte para múltiplas frenagens se diferente do código fonte para uma única parada apenas na seção da solução do sistema linear e no coeficiente de partição do calor. A seção é mostrada abaixo.

```
#MULTIPLAS FRENAGENS
```

```
gamma = 1/(1+ (kp*Sp/(Sd*(kp+h*p)))
```

```
#coeficiente de partição de calor
```

```
#SOLUCAO DO SISTEMA LINEAR
```

```
dt = 0.001
```

```
theta = 0.5 #discretizacao temporal
```

```
 #(0 = explicito , 0.5 = cranck-nicolson , 1 = implicito)
```

```
T = Fi
```

```
n = 100 #numero de paradas
```

```
tresf = 50 #tempo de resfriamento em s
```

```
ta = 4.3 #tempo de aceleracao ate a velocidade maxima em s
```

```
vmax = 15 #velocidade maxima do veiculo
```

```
Tparadas = np.zeros(n+1)
```

```
paradas = list(range(0,n+1))
```

```
for j in range (0,n+1):
```

```
    Tparadas [j] = np.max(T)
```

```
    for i in range (0,int(1/dt)+1):
```

```

tempo = i*(tb*dt)
if tempo == tb:
    for w in range (0,int(tresf/(ta*dt))+1):

        #MONTAGEM DA MATRIZ DE RIGIDEZ CONVECTIVA

        if (w*(ta*dt))<ta:
            tempo = w*(ta*dt)
        else:
            tempo = ta

v = vmax #velocidade maxima do veiculo
        #em m/s
omega = (v/Rr)*(tempo/ta) #velocidade
$angular do disco de freio em rad/s
Re1 = (omega/vis)*(2*((rp+rd)/2000))**2
#numero de Reynolds para conveccao
#na borda esquerda da malha 1 (jk)
Re2 = (omega/vis)*(2*((Rp+rp)/2000))**2
#numero de Reynolds para conveccao
#na borda esquerda da malha 2 (jk)
Re3 = (omega/vis)*(2*((Rd+Rp)/2000))**2
#numero de Reynolds para conveccao
#na borda esquerda da malha 3 (jk)
Re4 = (omega/vis)*(2*(Rd/1000))**2
#numero de Reynolds para conveccao
#na borda superior da malha 3 (ij)

if Re1 <= 10**6: #definicao do coeficiente
#de conveccao em W/m2C
    h1 = (0.7*kar/(2*((rp+rd)/2000)))*
        (Re1**0.55)

```

```

else:
    h1 = (0.4 * kar / (2 * ((rp + rd) / 2000))) *
        (Re1 ** 0.8)

if Re2 <= 10 ** 6: #definicao do coeficiente
#de conveccao em W/m2C
    h2 = (0.7 * kar / (2 * ((Rp + rp) / 2000))) *
        (Re2 ** 0.55)
else:
    h2 = (0.4 * kar / (2 * ((Rp + rp) / 2000))) *
        (Re2 ** 0.8)

if Re3 <= 10 ** 6: #definicao do coeficiente
#de conveccao em W/m2C
    h3 = (0.7 * kar / (2 * ((Rd + Rp) / 2000))) *
        (Re3 ** 0.55)
else:
    h3 = (0.4 * kar / (2 * ((Rd + Rp) / 2000))) *
        (Re3 ** 0.8)

if Re4 <= 10 ** 6: #definicao do coeficiente
#de conveccao em W/m2C
    h4 = (0.7 * kar / (2 * (Rd / 1000))) * (Re4 ** 0.55)
else:
    h4 = (0.4 * kar / (2 * (Rd / 1000))) * (Re4 ** 0.8)

if omega == 0: #conveccao natural em W/m2C
    h1 = 5
    h2 = 5
    h3 = 5
    h4 = 5

```

```

H1 = h1/kd #razao utilizada para
#simplificacaodos calculos: m-1 - borda
#esquerda da malha 1 (jk)
H2 = h2/kd #razao utilizada para
#simplificacao dos calculos: m-1 - borda
#esquerda da malha 2 (jk)
H3 = h3/kd #razao utilizada para
#simplificacao dos calculos: m-1 - borda
3superior da malha 3 (ij)
H4 = h4/kd #razao utilizada para
#simplificacao dos calculos: m-1 -
#borda superior da malha 3 (ij)

kh11 = np.dot((2*np.pi*dr1*H1/12),
[[0,0,0],[0,4,2],[0,2,4]]) #matriz rigidez
convectiva na borda esquerda da malha 1 (jk)
kh12 = np.dot((2*np.pi*dr2*H2/12),
[[0,0,0],[0,4,2],[0,2,4]]) #matriz rigidez
convectiva na borda esquerda da malha 2 (jk)
kh13 = np.dot((2*np.pi*dr3*H3/12),
[[0,0,0],[0,4,2],[0,2,4]]) #matriz rigidez
convectiva na borda esquerda da malha 3 (jk)
kh23 = np.dot((2*np.pi*dz*H4/12),
[[4*Rd/1000,2*Rd/1000,0],
[2*Rd/1000,4*Rd/1000,0],[0,0,0]])#matriz
#rigidez convectiva na borda superior
#da malha 3 (ij)

Kh11 = np.zeros((npoints,npoints))
#cria a matriz rigidez convectiva global
#na borda esquerda da malha 1(jk)
Kh12 = np.zeros((npoints,npoints))

```

```

#cria a matriz rigidez convectiva global
#na borda esquerda da malha 2 (jk)
Kh13 = np.zeros((npoints, npoints))
#cria a matriz rigidez convectiva global
#na borda esquerda da malha 3 (jk)
Kh23 = np.zeros((npoints, npoints))
#cria a matriz rigidez convectiva global
#na borda superior da malha 3 (ij)

nh11 = [] #cria os vetores dos elementos
convectivos na borda esquerda da malha 1
for i in range(0, ne1, 2*(nz-1)):
    nh11.append(i)

nh12 = [] #cria os vetores dos elementos
convectivos na borda esquerda da malha 2
for i in range(ne1, (ne1+ne2), 2*(nz-1)):
    nh12.append(i)

nh13 = [] #cria os vetores dos elementos
convectivos na borda esquerda da malha 3
for i in range((ne1+ne2), ne, 2*(nz-1)):
    nh13.append(i)

nh23 = [] #cria os vetores dos elementos
convectivos na borda superior
for i in range (ne-(2*nz-3), ne+1, 2):
    nh23.append(i)

for elem in nh11:
    for ilocal in range(0, 3):
        iglobal = int(IEN[elem, ilocal])

```

```

    for jlocal in range (0,3):
        jglobal = int(IEN[elem,jlocal])
        Kh11[iglobal,jglobal] =
            Kh11[iglobal,jglobal] +
            Rm[elem,0]*kh11[ilocal,jlocal]

for elem in nh12:
    for ilocal in range(0,3):
        iglobal = int(IEN[elem,ilocal])
        for jlocal in range (0,3):
            jglobal = int(IEN[elem,jlocal])
            Kh12[iglobal,jglobal] =
                Kh12[iglobal,jglobal] +
                Rm[elem,0]*kh12[ilocal,jlocal]

for elem in nh13:
    for ilocal in range(0,3):
        iglobal = int(IEN[elem,ilocal])
        for jlocal in range (0,3):
            jglobal = int(IEN[elem,jlocal])
            Kh13[iglobal,jglobal] =
                Kh13[iglobal,jglobal] +
                Rm[elem,0]*kh13[ilocal,jlocal]

for elem in nh23:
    for ilocal in range(0,3):
        iglobal = int(IEN[elem,ilocal])
        for jlocal in range (0,3):
            jglobal = int(IEN[elem,jlocal])
            Kh23[iglobal,jglobal] =
                Kh23[iglobal,jglobal] +
                kh23[ilocal,jlocal]

```

```

Kh = Kh11+Kh12+Kh13+Kh23 #matriz rigidez
                                #convectiva global

```

```

#MONTAGEM DA MATRIZ RIGIDEZ TOTAL

```

```

K = Kc + Kh

```

```

#MONTAGEM DO VETOR FORCA - CONVECCAO

```

```

fh11 = np.dot((2*np.pi*dr1*H1*Tf)/6
, [[0],[3],[3]])
#cria o vetor forca convectiva do elemento

```

```

na malha 1 na borda esquerda

```

```

fh12 = np.dot((2*np.pi*dr2*H2*Tf)/6,
[[0],[3],[3]])

```

```

#cria o vetor forca convectiva do elemento
na malha 2 na borda esquerda

```

```

fh13 = np.dot((2*np.pi*dr3*H3*Tf)/6,
[[0],[3],[3]])

```

```

#cria o vetor forca convectiva do elemento
na malha 3 na borda esquerda

```

```

fh23 = np.dot((2*np.pi*dz*H4*Tf)/6,
[[3*Rd/1000],[3*Rd/1000],[0]])

```

```

#cria o vetor forca convectiva do elemento
na malha 3 na borda superior

```

```

fh = np.zeros((npoints,1))

```

```

#cria o vetor forca convectiva

```

```

for elem in nh11:

```

```

    for jlocal in range(0,3):

```

```

        jglobal = int(IEN[elem, jlocal])
        fh[jglobal, 0] = fh[jglobal, 0] +
        Rm[elem, 0]*fh11[jlocal, 0]

for elem in nh12:
    for jlocal in range(0, 3):
        jglobal = int(IEN[elem, jlocal])
        fh[jglobal, 0] = fh[jglobal, 0] +
        Rm[elem, 0]*fh12[jlocal, 0]

for elem in nh13:
    for jlocal in range(0, 3):
        jglobal = int(IEN[elem, jlocal])
        fh[jglobal, 0] = fh[jglobal, 0] +
        Rm[elem, 0]*fh13[jlocal, 0]

for elem in nh23:
    for jlocal in range(0, 3):
        jglobal = int(IEN[elem, jlocal])
        fh[jglobal, 0] = fh[jglobal, 0] +
        fh23[jlocal, 0]

```

#MONTAGEM DA EQUACAO (ASSEMBLY)

```

A = M/dt + theta*K
    #matriz que compoe o tempo n+1
B = M/dt - (1-theta)*K
    #matriz que compoe o tempo n

b = np.dot(B,T) - fh #vetor solucao
T = np.linalg.solve(A,b) #solucao

```

else :

#MONTAGEM DA MATRIZ DE RIGIDEZ CONVECTIVA

$v = v_{max}$

#velocidade maxima do veiculo em m/s

$\omega = (v/Rr) * (1 - (tempo/tb))$

#velocidade angular

#do disco de freio em rad/s

$Re1 = (\omega/vis) * (2 * ((rp+rd)/2000)) ** 2$

#numero de Reynolds para conveccao

#na borda esquerda da malha 1 (jk)

$Re2 = (\omega/vis) * (2 * ((Rd+Rp)/2000)) ** 2$

#numero de Reynolds para conveccao

#na borda esquerda da malha 3 (jk)

$Re3 = (\omega/vis) * (2 * (Rd/1000)) ** 2$

#numero de Reynolds para conveccao

#na borda superior da malha 3 (ij)

if $Re1 \leq 10 ** 6$: *#definicao do coeficiente de conveccao em W/m2C*

$h1 = (0.7 * kar / (2 * ((rp+rd)/2000))) * (Re1 ** 0.55)$

else :

$h1 = (0.4 * kar / (2 * ((rp+rd)/1000))) * (Re1 ** 0.8)$

if $Re2 \leq 10 ** 6$: *#definicao do coeficiente de conveccao em W/m2C*

$h2 = (0.7 * kar / (2 * ((Rd+Rp)/2000))) * (Re2 ** 0.55)$

else :

$h2 = (0.4 * kar / (2 * ((Rd+Rp)/1000))) * (Re2 ** 0.8)$

(Re2**0.8)

if Re3 <= 10**6: *#definicao do coeficiente*
de conveccao em W/m2C

h3 = (0.7*kar/(2*(Rd/1000)))*(Re3**0.55)

else:

h3 = (0.4*kar/(2*(Rd/1000)))*(Re3**0.8)

if omega == 0: *#conveccao natural em W/m2C*

h1 = 5

h2 = 5

h3 = 5

H1 = h1/kd *#razao utilizada para*

#simplificacao dos calculos: m-1 - borda

#esquerda da malha 1 (jk)

H2 = h2/kd *#razao utilizada para*

#simplificacao dos calculos: m-1 - borda

#esquerda da malha 3 (jk)

H3 = h3/kd *#razao utilizada para*

#simplificacao dos calculos: m-1 - borda

#superior da malha 3 (ij)

kh11 = np.dot((2*np.pi*dr1*H1/12),

[[0,0,0],[0,4,2],[0,2,4]]) *#matriz rigidez*

#convectiva na borda esquerda da malha 1 (jk)

kh13 = np.dot((2*np.pi*dr3*H2/12),

[[0,0,0],[0,4,2],[0,2,4]]) *#matriz rigidez*

#convectiva na borda esquerda da malha 3 (jk)

kh23 = np.dot((2*np.pi*dz*H3/12),

[[4*Rd/1000,2*Rd/1000,0],

[2*Rd/1000,4*Rd/1000,0],[0,0,0]]) *#matriz rigidez*

convectiva na borda superior da malha 3 (ij)

```
Kh11 = np.zeros((npoints, npoints), dtype='float')
```

```
#cria a matriz rigidez convectiva global
```

```
na borda esquerda da malha 1(jk)
```

```
Kh13 = np.zeros((npoints, npoints), dtype='float')
```

```
#cria a matriz rigidez convectiva global
```

```
na borda esquerda da malha 3 (jk)
```

```
Kh23 = np.zeros((npoints, npoints), dtype='float')
```

```
#cria a matriz rigidez convectiva global
```

```
na borda superior da malha 3 (ij)
```

```
nh11 = [] #cria os vetores dos elementos
```

```
convectivos na borda esquerda da malha 1
```

```
for i in range(0, ne1, 2*(nz-1)):
```

```
    nh11.append(i)
```

```
nh13 = [] #cria os vetores dos elementos
```

```
convectivos na borda esquerda da malha 3
```

```
for i in range((ne1+ne2), ne, 2*(nz-1)):
```

```
    nh13.append(i)
```

```
nh23 = [] #cria os vetores dos elementos
```

```
convectivos na borda superior
```

```
for i in range (ne-(2*nz-3), ne+1, 2):
```

```
    nh23.append(i)
```

```
for elem in nh11:
```

```
    for ilocal in range(0, 3):
```

```
        iglobal = int(IEN[elem, ilocal])
```

```
        for jlocal in range (0, 3):
```

```
            jglobal = int(IEN[elem, jlocal])
```

```

        Kh11[iglobal , jglobal] =
        Kh11[iglobal , jglobal] +
        Rm[elem,0]*kh11[ilocal , jlocal]

for elem in nh13:
    for ilocal in range(0,3):
        iglobal = int(IEN[elem,ilocal])
        for jlocal in range (0,3):
            jglobal = int(IEN[elem,jlocal])
            Kh13[iglobal , jglobal] =
            Kh13[iglobal , jglobal] +
            Rm[elem,0]*kh13[ilocal , jlocal]

for elem in nh23:
    for ilocal in range(0,3):
        iglobal = int(IEN[elem,ilocal])
        for jlocal in range (0,3):
            jglobal = int(IEN[elem,jlocal])
            Kh23[iglobal , jglobal] =
            Kh23[iglobal , jglobal] +
            Rm[elem,0]*kh23[ilocal , jlocal]

Kh = Kh11+Kh13+Kh23
    #matriz rigidez convectiva global

#MONTAGEM DA MATRIZ RIGIDEZ TOTAL

K = Kc + Kh

#MONTAGEM DO VETOR FORCA - CONVECCAO

fh1 = np.dot((2*np.pi*dr1*H1*Tf)/6

```

```

,[[0],[3],[3]])
#cria o vetor forza convectiva do elemento
#na malha 1 na borda esquerda
fh13 = np.dot((2*np.pi*dr3*H2*Tf)/6,
[[0],[3],[3]])
#cria o vetor forza convectiva do elemento
#na malha 3 na borda esquerda
fh23 = np.dot((2*np.pi*dz*H3*Tf)/6,
[[3*Rd/1000],[3*Rd/1000],[0]])
#cria o vetor forza convectiva do elemento
#na malha 3 na borda superior

fh = np.zeros((npoints,1))
#cria o vetor forza convectiva

for elem in nh11:
    for jlocal in range(0,3):
        jglobal = int(IEN[elem,jlocal])
        fh[jglobal,0] = fh[jglobal,0] +
        Rm[elem,0]*fh1[jlocal,0]

for elem in nh13:
    for jlocal in range(0,3):
        jglobal = int(IEN[elem,jlocal])
        fh[jglobal,0] = fh[jglobal,0] +
        Rm[elem,0]*fh13[jlocal,0]

for elem in nh23:
    for jlocal in range(0,3):
        jglobal = int(IEN[elem,jlocal])
        fh[jglobal,0] = fh[jglobal,0] +
        Rm[elem,0]*fh23[jlocal,0]

```

```
#MONTAGEM DO VETOR FORCA - FLUXO DE CALOR
```

```
qi = (1/kd)*(phi/(2*np.pi))*gamma*mi*Pm*omega  
#fluxo de calo inicial em W/m2 (t=0)
```

```
fq2 = np.dot((2*np.pi*dr2*qi)/6,[[0],[3],[3]])
```

```
nh12 = [] #cria os vetores
```

```
#dos elementos de fluxona borda esquerda
```

```
#da malha 2
```

```
for i in range(ne1, ne1+ne2, 2*(nz-1)):
```

```
    nh12.append(i)
```

```
fq = np.zeros((npoints,1), dtype='float')
```

```
#cria o vetor forca por fluxo de calor
```

```
for elem in nh12:
```

```
    for jlocal in range(0,3):
```

```
        jglobal = int(IEN[elem, jlocal])
```

```
        fq[jglobal,0] = fq[jglobal, 0] +
```

```
        Rm[elem,0]*fq2[jlocal,0]*
```

```
        (R[jglobal]/1000)
```

```
#MONTAGEM DA EQUACAO (ASSEMBLY)
```

```
A = M/dt + theta*K
```

```
    #matriz que compoe o tempo n+1
```

```
B = M/dt - (1-theta)*K
```

```
    #matriz que compoe o tempo n
```

```
b = np.dot(B,T) + fh - fq #vetor solucao
```

```
T = np.linalg.solve(A,b) #solucao numerica
```

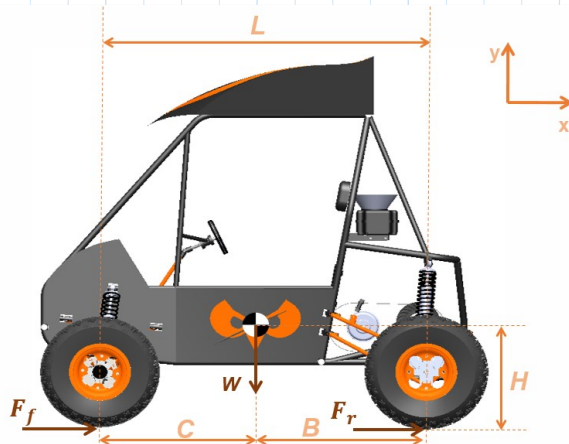
Apêndice C

Dimensionamento do Sistema de Freios

Este Apêndice fornece as informações necessárias do sistema de freio para o trabalho fornecidos pela Equipe Minerva Baja da Universidade Federal do Rio de Janeiro.

DCL, DADOS E FÓRMULAS

Obs.: Dados de input em vermelho



Distância entre eixos (L)

$$L := 1.4 \text{ m}$$

Altura do CG (H)

$$H := 0.53 \text{ m}$$

Massa (M)

$$M := 280 \text{ kg}$$

% de massa na parte traseira (x)

$$x := 0.6$$

Distância CG eixo traseiro/dianteiro (b/c)

$$c := x \cdot L = 0.84 \text{ m} \quad b := (1 - x) \cdot L = 0.56 \text{ m}$$

Peso máximo do veículo (W)

$$W := M \cdot g = (2.746 \cdot 10^3) \text{ N}$$

$$g = 9.807 \frac{\text{m}}{\text{s}^2}$$

Peso estático sobre o eixo dianteiro (W_{fs})

$$W_{fs} := M \cdot (1 - x) \cdot g = (1.098 \cdot 10^3) \text{ N}$$

Peso estático sobre o eixo traseiro (W_{rs})

$$W_{rs} := W - W_{fs} = (1.648 \cdot 10^3) \text{ N}$$

Aceleração máxima de uma freada (a_x), travando as rodas em uma distância ΔS , com velocidade inicial igual a V_0 e final igual a V ; desconsiderando o tempo de reação do piloto para acionar o pedal.

$$V := 0 \frac{\text{m}}{\text{s}}$$

$$V_0 := 9.5 \frac{\text{m}}{\text{s}}$$

$$\Delta S := 8 \text{ m}$$

$$a_x := \frac{V^2 - V_0^2}{2 \cdot \Delta S} = -5.641 \frac{\text{m}}{\text{s}^2}$$

$$a_{x'} := \frac{-a_x}{g} = 0.575$$

Transferência de carga dinâmica para o eixo dianteiro (W_f)

$$W_f := W \cdot \left(\frac{b}{L} + a_{x'} \cdot \frac{H}{L} \right) = (1.696 \cdot 10^3) \text{ N}$$

Transferência de carga dinâmica para o eixo traseiro (W_r)

$$W_r := W \cdot \left(\frac{c}{L} - a_{x'} \cdot \frac{H}{L} \right) = (1.05 \cdot 10^3) \text{ N}$$

DIMENSIONAMENTO DO SISTEMA DE FREIO

Coefficiente de atrito pneu-solo (μ), em solo de terra úmida

$$\mu := 0.65$$

TERRENO	COEFICIENTE DE ATRITO
ASFALTO	0,86
ASFALTO MOLHADO	0,76
RUA DE PARALELEPÍPEDO	0,75
TERRA	0,43 - 0,66
LAMA	0,31 - 0,39

Força aplicada na dianteira (Ff)
[Fat nos pneus dianteiros]

Força aplicada na traseira (Fr)
[Fat nos pneus traseiros]

$$F_f := W_f \cdot \mu = (1.103 \cdot 10^3) \text{ N}$$

$$F_r := W_r \cdot \mu = 682.247 \text{ N}$$

Raio efetivo da roda (R)

$$R := 0.267 \text{ m}$$

Momento aplicado na dianteira (Mf)

Momento aplicado na traseira (Mr)

$$M_f := F_f \cdot R = 294.384 \text{ J}$$

$$M_r := F_r \cdot R = 182.16 \text{ J}$$

Raio efetivo dos discos (Rd)

$$R_d := 0.0705 \text{ m}$$

Força aplicada nos discos dianteiros (Fdf)
[fat discos dianteiros]

Força aplicada nos discos traseiros (Fdr)
[fat disco traseiro]

$$F_{df} := \frac{M_f}{R_d} = (4.176 \cdot 10^3) \text{ N}$$

$$F_{dr} := \frac{M_r}{R_d} = (2.584 \cdot 10^3) \text{ N}$$

Coefficiente da atrito pastilha-disco (μ_d)

$$\mu_d := 0.4$$

Força aplicada nas pastilhas dianteiras (Ff')
[normal aos discos]

Força aplicada nas pastilhas traseiras (Fr')
[normal ao disco]

$$F_{f'} := \frac{F_{df}}{\mu_d} = (1.044 \cdot 10^4) \text{ N}$$

$$F_{r'} := \frac{F_{dr}}{\mu_d} = (6.46 \cdot 10^3) \text{ N}$$

Diâmetro do embolo da pinça (Wilwood SC2)

$$\phi_p := 0.030226 \text{ m}$$

Área do êmbolo da pinça (Wilwood SC2)

$$A := \pi \cdot \left(\frac{\phi_p}{2} \right)^2 = (7.175 \cdot 10^{-4}) \text{ m}^2$$

Nº de embolos/pinças na dianteira
(n_{ef})

$$n_{ef} := 2$$

Nº de embolos/pinças na traseira
(n_{er})

$$n_{er} := 1$$

Pressão no sistema dianteiro (ρ_f)

$$\rho_f := \frac{\frac{F_f}{2}}{A \cdot n_{ef}} = (3.637 \cdot 10^6) \text{ Pa}$$

Pressão no sistema traseiro (ρ_r)

$$\rho_r := \frac{\frac{F_r}{2}}{A \cdot n_{er}} = (4.501 \cdot 10^6) \text{ Pa}$$