



UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA

HEITOR AUGUSTO CARNEIRO DE ALMEIDA

**DESENVOLVIMENTO DE CÓDIGO EM LINGUAGEM C++ PARA
MODELAMENTO DO COMPORTAMENTO TERMODINÂMICO DE
PNEU UTILIZANDO O MÉTODO DOS ELEMENTOS FINITOS**

JOÃO PESSOA – PB

2018

HEITOR AUGUSTO CARNEIRO DE ALMEIDA

**DESENVOLVIMENTO DE CÓDIGO EM LINGUAGEM C++ PARA
MODELAMENTO DO COMPORTAMENTO TERMODINÂMICO DE
PNEU UTILIZANDO O MÉTODO DOS ELEMENTOS FINITOS**

Trabalho de Conclusão de Curso apresentado à banca
examinadora do Departamento de Engenharia Mecânica da
Universidade Federal da Paraíba como parte das exigências
à obtenção do título de Bacharel em Engenharia Mecânica.
Orientador: Prof. Dr. Marcelo Cavalcanti Rodrigues

JOÃO PESSOA – PB

2018

A447d Almeida, Heitor Augusto Carneiro de.

Desenvolvimento de código em linguagem C++ para modelamento do comportamento termodinâmico de pneu utilizando o Método dos Elementos Finitos / Heitor Augusto Carneiro de Almeida. - João Pessoa, 2018. 55 f. : il.

Orientação: Marcelo Cavalcanti Rodrigues.
Monografia (Graduação) - UFPB/CT.

1. Método de Elementos Finitos (MEF). 2. Modelamento de Pneu. 3. Análise Térmica-estrutural. 4. Intercambiabilidade. 5. Pneu. 6. C++. I. Rodrigues, Marcelo Cavalcanti. II. Título.

UFPB/BC

HEITOR AUGUSTO CARNEIRO DE ALMEIDA

**DESENVOLVIMENTO DE CÓDIGO EM LINGUAGEM C++ PARA
MODELAMENTO DO COMPORTAMENTO TERMODINÂMICO DE
PNEU UTILIZANDO O MÉTODO DOS ELEMENTOS FINITOS**

Trabalho de Conclusão de Curso – TCC, apresentado à banca examinadora do Departamento de Engenharia Mecânica da Universidade Federal da Paraíba como parte das exigências à obtenção do título de Bacharel em Engenharia Mecânica, tendo obtido o conceito **APROVADO** conforme a apreciação da banca examinadora:

João Pessoa, 07 de Novembro de 2018

BANCA EXAMINADORA

MARCELO CAVALCANTI RODRIGUES

Prof. Dr. Marcelo Cavalcanti Rodrigues

Orientador

Departamento de Engenharia Mecânica / Centro de Tecnologia / UFPB

Koje Daniel C. Mishina

Prof. Dr. Koje Daniel Vasconcelos Mishina

Departamento de Engenharia Mecânica / Centro de Tecnologia / UFPB

JOSÉ CARLOS DE LIMA JÚNIOR

Prof. Dr. José Carlos de Lima Junior

Departamento de Engenharia Mecânica / Centro de Tecnologia / UFPB

Dedico este trabalho à minha família que sempre esteve ao meu lado, apoiando e incentivando, tornando possível a conclusão desse curso.

AGRADECIMENTOS

Agradeço primeiramente à minha mãe, Maria de Lourdes Vieira, que me deu apoio incondicional durante toda minha jornada. Ao meu pai, José Carneiro de Almeida (in memorian), que sempre foi e será força e inspiração na minha vida. À minha namorada, Renata Pires, que sempre me incentivou e apoiou durante o curso. Também à minha família, que não mediu esforços para ajudar.

Ao Prof. Dr. Marcelo Cavalcanti, pela orientação, sugestões, suporte teórico e introdução da matéria.

Ao Prof. Dr.-Ing. Christian Oertel, pelo convite de colaborar em um projeto como o RMOD-K, pela sua orientação, suporte técnico e teórico.

Ao Ing. Jan Hempel, pelo suporte técnico e teórico e, principalmente, pelos ensinamentos no campo da computação e implementação de códigos de elementos finitos.

À Robert Bräunlich, técnico do Laboratório de Mecatrônica da Universidade Técnica de Brandenburg, pela prontidão e auxílio com os equipamentos e licenças de *software*.

Aos meus colegas e amigos, Tiago Fonseca, Thalles Emannuel, Genilton de França, João Vinicius, Gessé Paulo, Dener Delmiro, Lucas Periassu e José Rafael pela amizade, ajuda nas lutas diárias e por tornar os momentos mais agradáveis desde o início deste curso de graduação em Engenharia Mecânica.

“Uma caminhada de mil léguas começa com um simples primeiro passo”
(Lao Tsé)

“No meio da dificuldade encontra-se a oportunidade. ”
(Albert Einstein)

RESUMO

A análise do comportamento do pneu é essencial na avaliação do desempenho de um automóvel, visto que esse tem uma contribuição majoritária no comportamento dinâmico do veículo. Muitos modelos atualmente utilizados apresentam limitações no número de análises suportadas, por exemplo o modelo que representa o pneu como uma mola é restrito basicamente para análises modais. A análise utilizando o método dos elementos finitos pode englobar análises estáticas, dinâmicas e modais, além de ser capaz de representar características construtivas específicas como a inclinação das fibras metálicas no pneu, a qual varia dependendo da finalidade do mesmo. Sendo assim um modelo bastante robusto e acurado é proposto. Visando aumentar a acurácia do modelo de estruturas pneu em elementos finitos já em uso, foi desenvolvida uma aplicação para o modelamento termodinâmico do mesmo utilizando a linguagem C++. Para tal, foi necessário criar um programa que faz uso da malha gerada pelo modelo mecânico estrutural, adaptando-a para a análise térmica através da adição de camadas intermediárias de elementos hexaédricos, sem perder a intercambiabilidade com a malha utilizada na análise mecânica estrutural. Desta forma, é possível acoplar o modelo térmico com o mecânico estrutural, tornando o modelo mais acurado e robusto.

Palavras-Chave: Método de Elementos Finitos (MEF). Modelamento de Pneu. Análise Térmica-estrutural. Intercambiabilidade. Pneu. C++.

ABSTRACT

The tire behavior analysis is essential on the performance evaluation of an automobile, since it has a major contribution on the dynamic behavior of the vehicle. Many models now used are limited in number of supported analysis, for instance the model that represents the tire as a spring is basically restricted to modal analysis. The analysis using the finite element method can perform static, dynamic and frequency analysis, and it is also capable of representing constructive characteristics like the tire's metallic fibers inclination, which varies depending on the tire goal. Therefore, a robust and accurate model is proposed. Aiming to increase the accuracy of the model currently in use, one application for thermodynamic tire modeling was developed using C++ language. Therefore, it was necessary to develop a code that uses the mesh generated by the structural mechanical model, adapting it for the thermal analysis through intermediate hexahedral elements layers addition, such that the interchangeability with the original mesh is preserved. Finally, it is possible to couple the thermodynamic and structural mechanical models, making the model more robust and accurate.

Keywords: Finite Element Method (FEM). Tire Modeling. Thermo-structural analysis. Interchangeability. C++.

LISTA DE FIGURAS

Figura 1 – Diagrama com métodos utilizados para encontrar as equações do MEF	16
Figura 2 – Barra sob esforços axiais.....	17
Figura 3 – Discretização do domínio físico.....	20
Figura 4 – Numeração dos nós e das faces de um Elemento hexaédrico	30
Figura 5 – Mapeamento do domínio natural para o domínio físico	31
Figura 6 – Malha da estrutura trapezoidal utilizada	39
Figura 7 – Malha original do pneu	40
Figura 8 – Procedimento de adição de camadas.....	41
Figura 9 – Malha com camada de 0.1mm de espessura adicionada	42
Figura 10 – Malha da geometria trapezoidal com faces selecionadas com fluxo de calor (esquerda) e resultados (direita)	44
Figura 11 – Malha da geometria trapezoidal com faces selecionadas com convecção (esquerda) e resultados (direita)	45
Figura 12 – Malha da geometria trapezoidal com elemento selecionado como gerador de calor (esquerda) e resultados (direita)	45
Figura 13 – Resultado qualitativo da simulação no toróide, somente fluxo de calor (esquerda) e todas condições de contorno disponíveis (direita)	46
Figura 14 – Condições de contorno utilizadas na simulação das malhas de pneu	47
Figura 15 – Perfil de temperatura da simulação na malha original do pneu	48
Figura 16 – Resultado de temperaturas nodais na secção transversal do pneu	49
Figura 17 – Resultado de temperatura nodal na malha do pneu em apenas um conjunto de elementos adjacentes	49
Figura 18 – Condições de contorno para o teste transiente	52
Figura 19 – Corte transversal da malha com condições de contorno	53
Figura 20 – Corte transversal com zoom para visualização da camada metálica.....	53
Figura 21 – Perfil de temperatura esperado para simulação transiente especificada	54

LISTA DE TABELAS

Tabela 1 – Diferenças percentuais máximas nos testes realizados.....	50
--	----

LISTA DE SIGLAS E ABREVIATURAS

MEF – Método dos elementos finitos

RMOD-K – *Reifenmodellsystem Koppeln*

EDP – Equação diferencial Parcial

$u_{,x}$ – Derivada de u em relação a x ;

E – Modulo de Young;

f – Função Força Distribuída;

u – Função solução da EDP;

w – Função Peso

N^A – Função de interpolação em A

K – Matriz de rigidez

F – Vetor força

d – Vetor com resultado nodais

SUMÁRIO

1 INTRODUÇÃO	13
1.1 Objetivo geral	14
1.2 Objetivos específicos	14
2 REFERENCIAL TEÓRICO.....	15
2.1 Método de elementos finitos em problemas lineares 1D.....	17
2.2 Método de elementos finitos em problemas lineares 3D com variável escalar	28
2.2.1 Forma forte.....	28
2.2.2 Forma fraca	29
3 METODOLOGIA.....	36
3.1 Desenvolver um programa de elementos finitos para problemas termodinâmicos	36
3.2 Verificando resultados com software comercial.....	38
3.3 Adaptação do programa para ler as malhas do programa previamente desenvolvido	39
3.4 Adaptar a malha obtida para o problema termodinâmico (adição de camadas)	40
3.5 Verificação de resultados com software comercial	42
4 RESULTADOS E DISCUSSÃO	43
5 CONCLUSÃO.....	51
6 SUGESTÕES PARA TRABALHOS FUTUROS.....	52
REFERÊNCIAS	55

1 INTRODUÇÃO

Pneus são amplamente usados em diversas aplicações na indústria atual, como por exemplo na indústria automotiva, aeroespacial e máquinas de construção civil. Como a estabilidade e segurança desses sistemas dependem do design e confiança dos pneus, o modelamento destes e seus protótipos virtuais são uma importante etapa no processo de desenvolvimento do produto (PATEL *et al.*, 2016). Desta forma, o veículo virtual no processo de desenvolvimento deve providenciar um modelo gerado por tentativa e erro para todos os componentes (OERTEL *et al.*, 2001).

Conforme também explicado por Patel (2016), existem 3 principais técnicas no modelamento de pneu, cada uma com sua vantagem e desvantagem. Essas técnicas são: método analítico, método dos elementos discretos e método dos elementos finitos (MEF). De acordo com Oertel (2001), os testes a serem realizados por simulações pode variar desde dinâmica direcional até conforto em terrenos irregulares.

Comparando a técnica utilizando o método analítico e a técnica utilizando o método dos elementos finitos, nota-se que a o método analítico baseia-se em majoritariamente em dados experimentais e em modelos lineares, não levando em consideração deformações e inércias distribuídas do pneu (PATEL *et al.*, 2016). Por outro lado, a utilização do MEF no modelo do pneu para o uso do mesmo em um software de Dinâmica de Multicorpos requer um solver de elementos finitos rápido e robusto. Todavia, abordagens utilizando programas de elementos finitos de aplicações gerais sofrem com enormes custos computacionais em construção e montagem das matrizes de rigidez, independente da sua contribuição na rigidez total do pneu. Como reforça Oertel (2012), simulações de dinâmica veicular, em qualquer condição analisada, requer um modelo de pneu rápido e acurado.

Visando criar-se um modelo de pneu em Elementos finitos capaz de realizar simulações acuradas com baixo custo computacional, foi desenvolvido um modelo de pneu (RMOD-K) na Universidade Técnica de Brandenburgo. RMOD-K (*Reifenmodellsystem Koppeln*), que significa modelo de pneu acoplado ou acoplável, é um software desenvolvido pela Universidade Técnica de Brandenburgo em Parceria com empresas alemãs. Esse *software* tem como objetivo prever o comportamento do pneu de maneira eficiente para diversos casos, como por exemplo irregularidades na pista. O RMOD-K já foi utilizado por diversas empresas como *Audi*, *BMW*, *Porsche*, entre outras, sendo assim um *software* que já mostrou sua confiabilidade e

aplicabilidade. Este modelo de pneu reduz o custo computacional focando o modelamento em efeitos principais, como por exemplo aplicação da pré-carga na estrutura do pneu. Este modelo em elementos finitos também conta com a presença de elementos especiais para aplicação desejada, como por exemplo elementos para a cinta estabilizadora (*rebar elements*).

Todavia, esse modelo de pneu requer uma calibração, a qual irá definir os parâmetros que caracterizam o comportamento de cada pneu. A calibração dos modelos é feita através de experimentos padrões em laboratório e procedimentos de otimização de variáveis, o qual irá estimar os parâmetros que não são puramente físicos (OERTEL *et al.*, 2016).

Neste presente trabalho será desenvolvido um código que possibilitará a implementação de uma abordagem multifísica (termomecânica) na versão atual do RMOD-K.

1.1 Objetivo geral

O objetivo geral do presente trabalho é desenvolver uma aplicação capaz de calcular as temperaturas em um modelo de pneu em Elementos finitos. Para tal deve-se utilizar as informações da malha já existente do modelo mecânico estrutural, realizando-se adaptações a mesma, porém sem perder a intercambiabilidade entre as malhas.

1.2 Objetivos específicos

- a) Criar código capaz de resolver problemas termodinâmicos baseado na malha já existente;
- b) Realizar testes comparando os resultados com softwares comerciais, para verificar se a implementação e os cálculos estão corretos;
- c) Função para adaptação de malha para o problema termodinâmico, de tal forma que a nova malha mantenha intercambiabilidade com a antiga;
- d) Realizar testes novamente para checar a nova malha;
- e) Exportar resultados para o modulo mecânico estrutural do software;

2 REFERENCIAL TEÓRICO

A predição do comportamento do pneu depende de vários fatores, incluindo não-linearidades, anisotropia do material e geometria complexa do pneu. Desta forma, o desenvolvimento de um modelo analítico para algumas análises do pneu é uma tarefa impossível. Desta forma, o Método dos Elementos Finitos foi a forma escolhida para contornar esse problema, pois o mesmo é capaz de obter aproximações razoáveis e também provê um desenvolvimento prévio.

De acordo com Kim (2009), o Método dos Elementos Finitos é um método numérico que é utilizado para resolver equações diferenciais que caracterizam muitos problemas de engenharia. Kim (2009) também afirma que a popularidade desse método é dada ao fato de que o a aplicação deste procedimento em programas computacionais resulta em ferramentas bastante versáteis por natureza que podem resolver muitos problemas práticos com pouco treinamento.

O MEF irá obter um conjunto de equações a partir da Equação Diferencial Parcial que representa o problema físico. A solução dessas equações irá fornecer a solução obtida pelo método numérico, a qual pode coincidir com a solução analítica caso o problema tenha sido corretamente programado. Existem várias maneiras de se obter as equações do MEF, como mostra a figura abaixo.

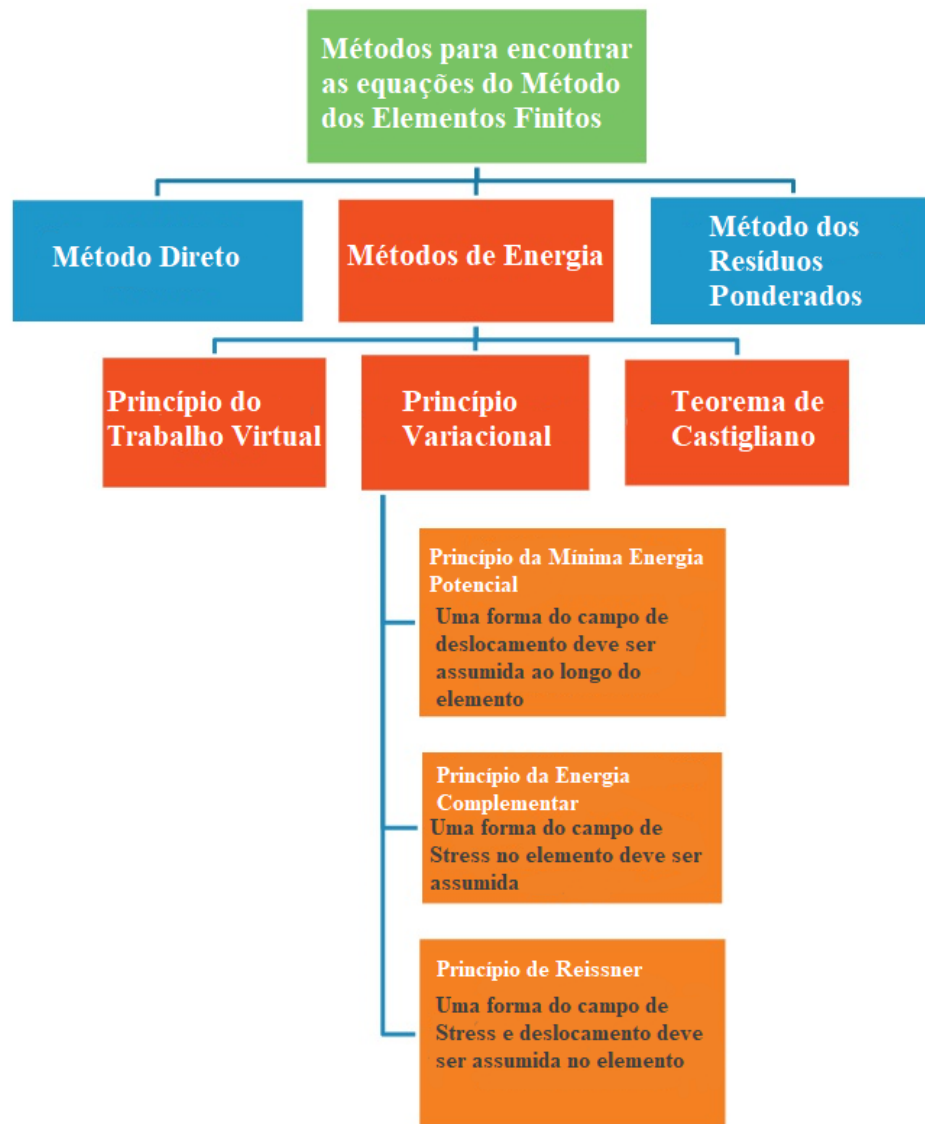


Figura 1 – Diagrama com métodos utilizados para encontrar as equações do MEF

Fonte: Pavlou (2015), adaptada pelo autor.

Cada abordagem tem suas limitações e vantagens, nesse trabalho o Método de Galerkin utilizando Elemento paramétricos será abordado. Essa abordagem necessita de uma quantidade considerável de tratamento matemático (Pavlou, 2015). Todavia, ao longo do desenvolvimento da mesma, algumas propriedades importantes do MEF podem ser analisadas. Portanto, neste trabalho será desenvolvido as equações para um Problema Elástico Estrutural 1D em regime permanente, para que alguns detalhes do método sejam analisados, como também para um Problema Termodinâmico 3D em Regime permanente, o qual foi utilizado no código desenvolvido ao final do trabalho.

2.1 Método de elementos finitos em problemas lineares 1D

Problemas lineares 1D podem ser representados como uma equação linear elíptica. Essas equações podem representar:

- Condução de calor unidimensional em regime permanente
- Difusão de massa unidimensional em regime permanente
- Elasticidade unidimensional em regime permanente

Considere o problema de elasticidade unidimensional em regime permanente mostrado na figura 2.

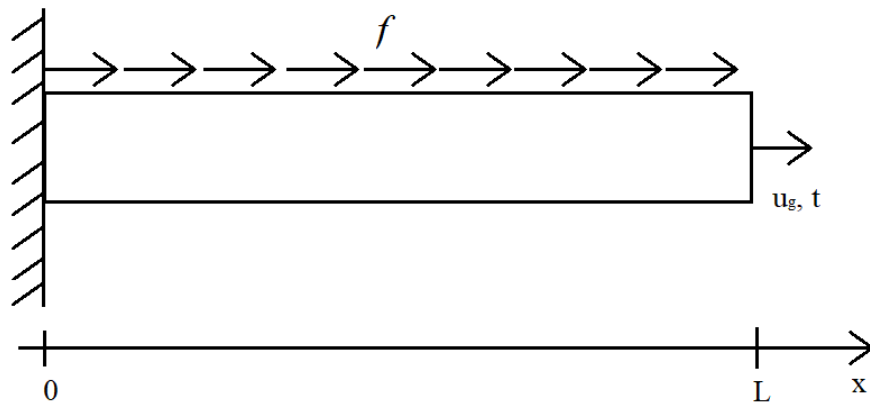


Figura 2 – Barra sob esforços axiais

Fonte: Própria do autor (2018).

A Equação Diferencial Parcial (EDP) que representa esse problema, a qual pode ser obtida a partir de livros de mecânica estrutural, é a seguinte:

$$E \frac{d^2 u}{dx^2} + f(x) = 0 \text{ em } (0, L) \quad (1)$$

Onde f é a força axial distribuída ao longo do corpo e u é o deslocamento, que neste problema é um escalar.

Utilizando a relação constitutiva, $\sigma = E \frac{du}{dx} = Eu_{,x}$, (onde E representa o Modulo de Young) pode-se representar a EDP como está representado abaixo:

$$\frac{d\sigma}{dx} + f = 0 \quad (2)$$

Para este problema, as condições de contorno podem ser:

- Dirichlet-Dirichlet

$$\begin{cases} u_0 = u(0) \\ u_g = u(L) \end{cases}$$

- Dirichlet-Neumann

$$\begin{cases} u_0 = u(0) \\ t = \sigma(L) \end{cases}$$

A condição de contorno Dirichlet, ou condição de contorno essencial, é a condição aplicada na campo primário que a EDP está sendo resolvida, como por exemplo neste caso o campo primário é o deslocamento u . A condição de contorno Neumann ou condição de contorno natural, por sua vez, é aplicada nas derivadas do campo primário. Note que para um problema que contém unicamente condições de contorno do tipo Neumann existe mais de uma solução que satisfaz as condições de contorno da EDP.

Considerando essas informações, o problema a ser resolvido pelo Método dos elementos finitos (MEF) pode ser escrito da seguinte forma:

Encontre $u(x): (0, L) \rightarrow \mathbb{R}$, dada as condições de contorno, função da força distribuída f e a relação constitutiva, $\sigma = Eu_{,x}$, tal que

$$\frac{d\sigma}{dx} + f = 0 \text{ em } (0, L) \quad (3)$$

Portanto, para um problema com condições de contorno Dirichlet-Neumann, $u(x)$ deve ser tal que,

$$u \in \mathbb{S} = \{u | u(0) = u_0\} \quad (4)$$

Já para problema com ambas condições de contorno do tipo Dirichlet, $u(x)$ deve ser tal que,

$$u \in \mathbb{S} = \{u | u(0) = u_0, u(L) = u_g\} \quad (5)$$

Onde $\mathbb{S}$ é o espaço de funções que contém a solução exata, u . Pode-se notar que essa formulação envolve duas derivadas da função solução. Logo ela requer fortes condições de suavidade na solução. Portanto, essa formulação é chamada de Forma Forte. Como essa condição algumas vezes não é possível de ser obtida, a formulação chamada de Forma Fraca foi desenvolvida para contornar esse problema. A Forma fraca pode ser obtida multiplicando ambos os lados da Forma forte por uma função peso $w(x)$, tal que

$$\forall w \in \mathbb{V} = \{w | w(0) = 0\} \quad (6)$$

Pode-se obter a Forma fraca, através de integração por partes e aplicando a condição de contorno Dirichlet homogênea em w , $w(0) = 0$. A formula fraca para um problema Dirichlet-Neumann é escrita abaixo:

$$\int_0^L w_{,x} \sigma A dx = \int_0^L w f A dx + w(L) t A \quad (7)$$

Sabendo que A é área da seção perpendicular a x , e que esta foi multiplicada de ambos os lados da equação para dar sentido físico para a formulação. Pode-se demonstrar que a Forma fraca e a Forma forte são equivalentes, pois, a Forma forte implica na forma fraca e vice-versa. Portanto, para a formulação que utiliza um espaço de funções com infinitas dimensões a seguinte afirmação é verdadeira:

$$\text{Forma forte} \Leftrightarrow \text{Forma fraca} \quad (8)$$

Como computadores não são capazes de resolver problemas em espaços de funções com infinitas dimensões, uma aproximação da solução foi desenvolvida. Essa abordagem utiliza espaços de funções com um número finito de dimensões. Portanto, utilizando o espaço de funções polinomiais $\mathbb{P}$ por exemplo, temos:

$$\mathbb{S}, \mathbb{V} \in \mathbb{P}^n, n = 0, 1, 2, \dots, k$$

Onde k é a dimensão (ou ordem) das funções base. Fazendo isso, a solução e a função peso ficam limitadas pelo espaço de função. Logo, tem-se para funções base lineares $k = 1$, funções quadráticas $k = 2$, e assim por diante. Portanto, a formulação que utiliza espaço de funções com de ordem finita, a qual pode ser chamada de Formulação fraca de Galerkin, é escrita da seguinte maneira:

Encontre $u^h \in \mathbb{S}^h = \{u^h \in \mathbb{H}^1(0, L) | u^h(0) = u_0\}$ tal que

$$\int_0^L w^h{}_{,x} \sigma A dx = \int_0^L w^h f A dx + w^h(L) t A \quad (9)$$

Onde, a solução experimental u^h e a função peso experimental w^h são tais que,

$$u^h \in \mathbb{S}^h \subset \mathbb{S}$$

$$\forall w^h \in \mathbb{V}^h \subset \mathbb{V}$$

$$\mathbb{V}^h = \{w^h \in \mathbb{H}^1(0, L) | w^h(0) = 0\} \quad (10)$$

Nesta formulação, $\mathbb{H}^1$ representa o espaço das funções de Hilbert. Este espaço é usado para aplicar certo controle sobre a solução. Para que uma função, $v(x)$, esteja no espaço de Hilbert a mesma deve obedecer a condição abaixo:

$$v(x) \in \mathbb{H}^1(0, L) \Leftrightarrow \int_0^L (v^2 + L^2 v^2{}_{,x}) dx < \infty \quad (11)$$

Note que L^2 foi introduzido para que as operações de soma sejam coerentes dimensionalmente. Em geral, L será a dimensão primitiva, para o problema que está sendo resolvido por exemplo, L representa comprimento.

Agora, para obter u^h, w^h deve-se dividir o domínio $(0, L)$ em Elementos Finitos, os quais são subdomínios desacoplados. O particionamento está representado na figura 3:

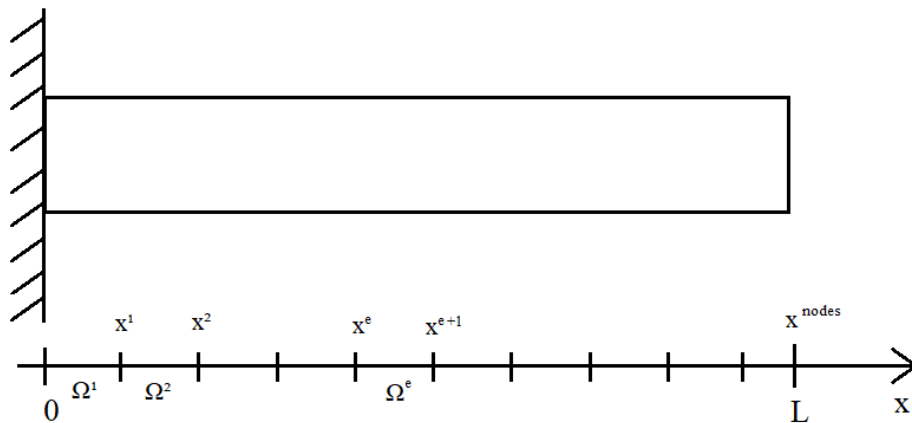


Figura 3 – Discretização do domínio físico

Fonte: Própria do autor (2018).

Agora a integral (9) pode ser escrita da seguinte forma:

$$\sum_{e=1}^{N^{el}} \int_{\Omega^e} w^h_{,x} \sigma^h A dx = \sum_{e=1}^{N^{el}} \int_{\Omega^e} w^h f A dx + w^h(L) t A \quad (12)$$

Visando representar as funções peso, w^h , e a solução experimental u^h , sobre os elementos, define-se funções base locais em Ω^e . Neste exemplo, onde cada elemento tem apenas dois nós, essas funções bases serão representadas funções base polinomiais lineares. Portanto, a solução experimental no elemento e será representada da seguinte maneira:

$$u_e^h(x) = \sum_{A=1}^{N^{ne}} N^A(x) d_e^A = N^1(x) d_e^1 + N^2(x) d_e^2 \quad (13)$$

Onde $N^A(x)$ é a função base, ou função de interpolação, do nó A e d_e^A é o grau de Liberdade do nó A . De forma similar para $w_e^h(x)$, escreve-se:

$$w_e^h(x) = \sum_{A=1}^{N^{ne}} N^A(x) c_e^A = N^1(x) c_e^1 + N^2(x) c_e^1 \quad (14)$$

Onde c_e^A é o grau de liberdade da função peso.

As funções base, N^A , são definidas em um domínio bi-unitário usando a expressão polinomial de Lagrange, representada na equação (15), para funções lineares. Polinômios de Lagrange também serão utilizados para elementos de ordem mais elevada. A razão de utilizar bases polinomiais se dá ao fato de que a Quadratura Gaussiana será utilizada para calcular as integrais no domínio bi-unitário. A Quadratura Gaussiana é usada pois esta retorna o valor exato da integral de polinômios, caso o número correto de pontos de Quadratura seja utilizado. Para uma base linear de um elemento 1D estão representadas nas funções (16) e (17):

$$N^A(\xi) = \frac{\prod_{\substack{B=1 \\ B \neq A}}^{N^{ne}} (\xi - \xi^B)}{\prod_{\substack{B=1 \\ B \neq A}}^{N^{ne}} (\xi^A - \xi^B)} \quad (15)$$

$$N^1(\xi) = \frac{1 - \xi}{2} \quad (16)$$

$$N^2(\xi) = \frac{1 + \xi}{2} \quad (17)$$

Vale também ressaltar que os polinômios de Lagrange também satisfazem a propriedade Kronecker Delta que é a seguinte:

$$N^A(\xi^B) = \delta_{AB} \quad (18)$$

Essa propriedade é crucial para esta formulação de elementos finitos, pois ela assegura que os graus de liberdade nodais (d_e^1, d_e^2) da solução são realmente esses valores nas posições nodais. Essa propriedade também providencia a propriedade de suporte local, a qual afirma que um grau de liberdade (nó) só irá influenciar a solução nos elementos adjacentes a ele. Esta propriedade é particularmente importante para a análise de singularidades. Outra propriedade é que a soma de todas as funções base no domínio bi-unitário é igual a unidade em todo o domínio bi-unitário. Como por exemplo, no problema que está a ser devolvido tem-se:

$$N^1(\xi) + N^2(\xi) = \frac{1 - \xi}{2} + \frac{1 + \xi}{2} = 1 \quad (19)$$

Essa propriedade permite a representação de constantes no domínio bi-unitário e, consequentemente, no domínio físico. Como as funções base são definidas no domínio natural (bi-unitário), é necessário definir um mapeamento do domínio natural para domínio físico. Para proceder com esse mapeamento as mesmas funções base serão usadas para u_e^h e w_e^h , o que constitui a formulação isoparamétrica. Este mapeamento é tal que:

$$x_e(\xi) = \sum_{A=1}^{Nne} (N^A(\xi)) x_e^A \quad (20)$$

Onde Nne representa o numero de nós no elemento e o índice e representa o elemento analisado. Portanto, para o problema que está sendo resolvido tem-se:

$$N^A(x) = N^A(x(\xi)) = N^A(\xi), \quad A = 1, 2 \quad (21)$$

Utilizando a equação (12), lembrada abaixo:

$$\sum_{e=1}^{N^{el}} \int_{\Omega^e} w^h_{,x} \sigma^h A dx = \sum_{e=1}^{N^{el}} \int_{\Omega^e} w^h f A dx + w^h(L) t A$$

Onde, N^{el} representa o número de elementos. Pode-se notar que as derivadas $w^h_{,x}$ e $u^h_{,x}$ (note também que $\sigma^h = u^h_{,x}$) devem ser calculadas nessa formulação. As derivadas calculadas podem ser encontradas a seguir:

$$u^h_{,x} = \sum_{A=1}^{Nne} N^A_{,\xi}(x) \xi_{,x} d_e^A \quad (22)$$

$$w^h_{,x} = \sum_{A=1}^{Nne} N^A_{,\xi}(x) \xi_{,x} c_e^A \quad (23)$$

Note que a regra da cadeia deve ser aplicada, pois $N^A(x) = N^A(x(\xi))$ e o termo $N^A_{,\xi}$ pode ser facilmente obtido. Por outro lado, $\xi_{,x}$ deve ser calculado através de um mapeamento do domínio bi-unitário para o domínio físico. Para o problema 3D, o qual será explicado mais adiante, esse termo será obtido utilizando a matriz Jacobiana. Calculando a derivada do domínio físico em relação ao domínio natural, tem-se:

$$x_{,\xi} = \sum_{A=1}^{Nne} (N^A(\xi))_{,\xi} x_e^A = N^1_{,\xi} x_e^1 + N^2_{,\xi} x_e^2 \quad (24)$$

Substituindo as bases, que podem ser encontradas nas equações (16) e (17), e aplicando a mudança de numeração do referencial local para o global através da relação $x_e^A = x^{e+A-1}$, tem-se:

$$x_{,\xi} = \frac{x^{e+1} - x^e}{2} = \frac{h^e}{2}$$

Portanto, nota-se que $\frac{h^e}{2}$ é o fator de escala, que representa o comprimento do elemento. É importante perceber que o fator de escala é uma constante pois o problema está utilizando uma função de interpolação linear para a representação da geometria. Finalmente, o mapeamento isoparamétrico é invertível pois ele é um mapa de um para um, logo pode-se obter a seguinte relação:

$$\xi_{,x} = \frac{1}{x_{,\xi}} = \frac{2}{h^e} \quad (25)$$

Aplicando esses resultados na equações (22) e (23), tem-se:

$$u^h_{,x} = \sum_{A=1}^{Nne} N^A_{,\xi}(x) \frac{2}{h^e} d_e^A \quad (26)$$

$$w^h_{,x} = \sum_{A=1}^{Nne} N^A_{,\xi}(x) \frac{2}{h^e} c_e^A \quad (27)$$

$$dx = \frac{dx}{d\xi} d\xi = \frac{h^e}{2} d\xi \quad (28)$$

Agora, as integrais presentes na equação (12) podem ser escritas da seguinte maneira:

$$\int_{\Omega^e} w^h_{,x} EA u^h_{,x} dx = \dots \quad (29)$$

$$\int_{\Omega^\xi} \left(\sum_{A=1}^{Nne} N^A_{,\xi}(x) \frac{2}{h^e} c_e^A \right) EA \left(\sum_{B=1}^{Nne} N^B_{,\xi}(x) \frac{2}{h^e} d_e^B \right) \frac{h^e}{2} d\xi$$

$$\int_{\Omega^e} w^h f(x(\xi)) A dx = \int_{\Omega^\xi} \left(\sum_{A=1}^{Nne} N^A(x) c_e^A \right) f(x(\xi)) A \frac{h^e}{2} d\xi \quad (30)$$

As quais podem ser simplificadas:

$$\int_{\Omega^\xi} \left(\sum_{A=1}^{Nne} N^A_{,\xi}(x) \frac{2}{h^e} c_e^A \right) EA \left(\sum_{B=1}^{Nne} N^B_{,\xi}(x) d_e^B \right) d\xi \quad (31)$$

$$\int_{\Omega^\xi} \left(\sum_{A=1}^{Nne} N^A(x) c_e^A \right) \frac{h^e}{2} f A d\xi \quad (32)$$

Considerando um elemento genérico Ω^e , e lembrando c_e^A que d_e^A são independentes de ξ , podemos escrever as duas integrais da seguinte maneira:

$$\int_{\Omega^\xi} \left(\sum_{A=1}^{Nne} N^A_{,\xi}(x) \frac{2}{h^e} c_e^A \right) EA \left(\sum_{B=1}^{Nne} N^B_{,\xi}(x) d_e^B \right) d\xi = \dots \quad (33)$$

$$\sum_{A,B=1}^{Nne} c_e^A \int_{\Omega^\xi} \left(N^A_{,\xi} \frac{2EA}{h^e} c_e^A N^B_{,\xi} d\xi \right) d_e^B$$

$$\int_{\Omega^\xi} \left(\sum_{A=1}^{Nne} N^A(x) c_e^A \right) \frac{h^e}{2} f A d\xi = \sum_{A=1}^{Nne} c_e^A \int_{\Omega^\xi} \left(N^A \frac{A h^e}{2} d\xi \right) \quad (34)$$

Agora, utilizando a notação matricial para eliminar $\sum_A$ e $\sum_B$ e considerando E e A uniformes ao longo de Ω^e , obtemos a seguinte equação:

$$\sum_{A,B=1}^{Nne} c_e^A \int_{\Omega^\xi} \left(N^A_{,\xi} \frac{2EA}{h^e} c_e^A N^B_{,\xi} d\xi \right) d_e^B = \dots$$

$$< c_e^1 c_e^2 > \frac{2EA}{h^e} \left(\int_{\Omega^\xi} \begin{bmatrix} N^1_{,\xi} & N^1_{,\xi} & N^2_{,\xi} & N^1_{,\xi} \\ N^1_{,\xi} & N^2_{,\xi} & N^2_{,\xi} & N^2_{,\xi} \end{bmatrix} d\xi \right) \begin{Bmatrix} d_e^1 \\ d_e^2 \end{Bmatrix} \quad (35)$$

Lembrando das bases utilizadas nesse problema, (16) e (17), podemos calcular suas derivadas em relação a ξ :

$$N^1(\xi) = \frac{1-\xi}{2} \Rightarrow N^1_{,\xi}(\xi) = -\frac{1}{2} \quad (36)$$

$$N^2(\xi) = \frac{1+\xi}{2} \Rightarrow N^2_{,\xi}(\xi) = \frac{1}{2} \quad (37)$$

Sabendo disso, pode-se substituir os valores na matriz e calcular a mesma integral em relação a ξ . Fazendo esse procedimento, e notando o valor da integral da equação (38), obtém-se:

$$\int_{\Omega^\xi} d\xi = \int_{-1}^1 d\xi = 2 \quad (38)$$

$$\int_{\Omega^e} w^h_{,x} EA u^h_{,x} dx = < c_e^1 c_e^2 > \frac{EA}{h^e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} d_e^1 \\ d_e^2 \end{Bmatrix} \quad (39)$$

De maneira similar, pode-se calcular a integral que contém a função da força distribuída f .

$$\sum_{A=1}^{Nne} c_e^A \int_{\Omega^\xi} \left(N^A \frac{Ah^e}{2} d\xi \right) = \langle c_e^1 c_e^2 \rangle \frac{fAh^e}{2} \int_{-1}^1 \begin{Bmatrix} N^1 \\ N^2 \end{Bmatrix} d\xi \quad (40)$$

$$\int_{\Omega^e} w^h f A dx = \langle c_e^1 c_e^2 \rangle \frac{fAh^e}{2} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \quad (41)$$

Substituindo essas equivalências na equação (12), obtém-se:

$$\sum_{e=1}^{Nel} \left(\langle c_e^1 c_e^2 \rangle \frac{EA}{h^e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} d_e^1 \\ d_e^2 \end{Bmatrix} \right) = \sum_{e=1}^{Nel} \left(\langle c_e^1 c_e^2 \rangle \frac{fAh^e}{2} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \right) + c_{Nel}^2 tA \quad (42)$$

Note que $w^h(L) = c_{Nel}^2$, devido a propriedade Kronecker delta, e que $\frac{EA}{h^e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$, é K_e , a qual representa a matriz de rigidez do elemento e .

Agora, faz-se necessário o uso da numeração dos nós no sistema global, para que seja possível realizar a montagem das matrizes, e assim aplicar as condições de contorno do problema. Para a conversão do Sistema de numeração local para o global, utilizam-se as seguintes formulas que são válidas no problema 1D com base linear:

$$d_e^A = d_{e+A-1} \quad (43)$$

$$c_e^A = c_{e+A-1} \quad (44)$$

Fazendo a conversão dos índices a acoplando as matrizes em uma única matriz, tem-se:

$$\begin{aligned} & \langle c_1 c_2 c_3 \dots c_{Nel} c_{Nel+1} \rangle \dots \\ & EA \begin{bmatrix} \frac{1}{h^1} & -\frac{1}{h^1} & & & \\ -\frac{1}{h^1} & \frac{1}{h^1} + \frac{1}{h^2} & -\frac{1}{h^2} & & \\ & -\frac{1}{h^2} & \frac{1}{h^2} + \frac{1}{h^3} & -\frac{1}{h^3} & \\ & & \ddots & \ddots & \\ & & -\frac{1}{h^3} & \frac{1}{h^{Nel-1}} + \frac{1}{h^{Nel}} & -\frac{1}{h^{Nel}} \\ & & & -\frac{1}{h^{Nel}} & \frac{1}{h^{Nel}} \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_{nel} \\ d_{nel+1} \end{Bmatrix} = \dots \end{aligned}$$

$$< c_1 \ c_2 \ c_3 \ \dots \ c_{Nel} \ c_{Nel+1} > \frac{fA}{2} \begin{Bmatrix} h^1 \\ h^1 + h^2 \\ h^2 + \dots \\ \vdots \\ h^{Nel-1} + h^{Nel} \\ h^{Nel} \end{Bmatrix} + < c_1 \ c_2 \ c_3 \ \dots \ c_{Nel} \ c_{Nel+1} > \begin{Bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ tA \end{Bmatrix}$$

Aplicando a condição de contorno homogênea do tipo Dirichlet em w , que é $c_1 = 0$, pode-se eliminar a primeira linha da matriz de rigidez completa e também da matriz de forças. Como o grau de liberdade d_1 é conhecido, a primeira coluna dessa matriz de rigidez também pode ser eliminada, passando assim para o lado direito da equação. Portanto a equação, com a condições de contorno já aplicadas, é a seguinte:

$$< c_2 \ c_3 \ \dots \ c_{Nel} \ c_{Nel+1} > EA \dots$$

$$\begin{bmatrix} \frac{1}{h^1} + \frac{1}{h^2} & -\frac{1}{h^2} & & & \\ -\frac{1}{h^2} & \frac{1}{h^2} + \frac{1}{h^3} & -\frac{1}{h^3} & & \\ & & \ddots & & \\ & -\frac{1}{h^3} & & \frac{1}{h^{Nel-1}} + \frac{1}{h^{Nel}} & -\frac{1}{h^{Nel}} \\ & & & -\frac{1}{h^{Nel}} & \frac{1}{h^{Nel}} \end{bmatrix} \begin{Bmatrix} d_2 \\ d_3 \\ \vdots \\ d_{nel} \\ d_{nel+1} \end{Bmatrix} = \dots$$

$$< c_2 \ c_3 \ \dots \ c_{Nel} \ c_{Nel+1} > \left(\frac{fA}{2} \begin{Bmatrix} h^1 + h^2 \\ h^2 + \dots \\ \vdots \\ h^{Nel-1} + h^{Nel} \\ h^{Nel} \end{Bmatrix} + \begin{Bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ tA \end{Bmatrix} - \frac{EA}{h_1} \begin{Bmatrix} u_0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{Bmatrix} \right)$$

$$\Rightarrow c^T K d = c^T F \quad (45)$$

Como essa equação deve ser válida $\forall w^h \in \mathbb{V}^h \Rightarrow \forall c \in \mathbb{R}^{Nel}$

$$\Rightarrow K d = F \quad (46)$$

A equação acima é a mais conhecida quando se trata de elementos finitos, e ela também é a equação matricial que corresponde ao problema. Sua solução é a seguinte

$$d = K^{-1}F \quad (47)$$

$$u^h = \sum_{A=1}^{Ndof} N^A d_e^A \quad (48)$$

2.2 Método de elementos finitos em problemas lineares 3D com variável escalar

A EDP resolvida para esse problema pode representar os seguintes problemas:

- Condução de calor 3D em regime estacionário
- Difusão de massa 3D em regime estacionário

O primeiro problema será abordado nesse trabalho, pois o mesmo será utilizado na análise termodinâmica do pneu. Seguindo a mesma sequência do procedimento utilizado para obter as equações para o caso da formulação 1D, inicialmente escreve-se a Forma Forte. Esta recebe esse nome pois requer fortes condições de suavidade na sua solução, devido a quantidade de derivadas na mesma. Em seguida, uma formulação com somente uma única derivada é encontrada através de procedimentos matemáticos, como está não precisa de condições tão rígidas de suavidade, a mesma é chamada de Forma Fraca.

2.2.1 Forma forte

Encontre u , dados $f(\underline{x})$, u_g , j_n , e a relação constitutiva $j_i = K_{ij}u_{,j}$ ($i, j = 1, 2, 3$), que é a Lei da Condutividade de Fourier em notação por coordenadas, K é um tensor, e função distribuída f representa a geração de calor. Tal que

$$-j_{i,i} = f \text{ em } \Omega \quad (49)$$

As condições de contorno, Dirichlet e Neumann, são as seguintes:

$$u = u_g(\underline{x}) \text{ em } \delta\Omega_u \quad (50)$$

$$-\underline{j} \cdot \underline{n} = j_n \text{ em } \delta\Omega_j \quad (51)$$

Onde $\underline{j}$ é o vetor que representar o fluxo de calor (quantidade de calor cruzando perpendicularmente em uma unidade de área por uma unidade de tempo), $\underline{x}$ é o vetor posição e ambos estão no espaço $\mathbb{R}^3$. Finalmente, o vetor $\underline{n}$ é o vetor normal unitário que aponta para fora da superfície $\delta\Omega_j$.

A forma forte pode ser escrita em notação direta, a qual representa todas as operações como operações de matrizes. Essa formulação é escrita da seguinte forma:

Encontre u dados u_g, j_n, f , a relação constitutiva em notação direta, $\underline{j} = -\underline{K}\underline{\nabla}u$, tal que:

$$\underline{K}\nabla^2 u = f \text{ em } \Omega \quad (52)$$

Onde, $\underline{\nabla}$ representa o operador gradiente especial e ∇^2 representa o operador Laplaciano.

Com as seguintes condições de contorno:

$$u = u_g \text{ em } \delta\Omega_u \quad (53)$$

$$\underline{K}\underline{\nabla}u \cdot \underline{n} = j_n \text{ em } \delta\Omega_j \quad (54)$$

Note que K representa a condutividade inerente as propriedades do material, para um material isotrópico por exemplo tem-se:

$$K_{ik} = k\delta_{ij} \quad (55)$$

De uma maneira análoga a desenvolvida para os elementos 1D, deve-se formular a Forma fraca para elementos 3D.

2.2.2 Forma fraca

Encontre $u \in \mathbb{S} = \{u | u = u_g \text{ em } \delta\Omega_u\}$ dados u_g, j_n, f e a relação constitutiva, $j_i = -K_{ij}u_{,j}$,

Tal que $\forall w \in \mathbb{V} = \{w | w = 0 \text{ em } \delta\Omega_u\}$

$$\int_{\Omega} w_{,i} j_i dV = \int_{\Omega} w f dV - \int_{\delta\Omega_j} w j_n dS \quad (56)$$

Onde dV é o volume do elemento e dS é a área do elemento. Vale ressaltar que, de uma maneira similar a resolvida para uma dimensão, pode-se provar que:

$$\text{Forma Forte} \Leftrightarrow \text{Forma Fraca}$$

Continuando a formulação de uma maneira análoga, tem-se a utilização de um espaço de funções finito, e assim uma solução experimental u^h é utilizada como aproximação do problema. Logo, o problema fica escrito da seguinte maneira:

Encontre $u^h \in \mathbb{S}^h \subset \mathbb{S}$, onde $\mathbb{S}^h = \{u^h \in \mathbb{H}^1(\Omega) | u^h = u_g \text{ em } \delta\Omega_u\}$, dados: $u_g, j_n, f, j_i^h = -K_{ij} u_{,j}^h$

Tal que $\forall w^h \in \mathbb{V}^h \subset V$, $\mathbb{V}^h = \{w^h \in \mathbb{H}^1(\Omega) | w^h = 0 \text{ em } \delta\Omega_u\}$

$$\int_{\Omega} w_{,i}^h j_i^h dV = \int_{\Omega} w^h f dV - \int_{\delta\Omega_j} w^h j_n dS \quad (57)$$

Agora, deve-se discretizar o domínio em elementos Ω^e , $e = 1, 2, \dots, N_{el}$. Para esse problema serão considerados elementos hexaédricos, também chamados de *Hex8* ou *8-Node Brick* pois, além de ser atualmente utilizado no programa de pneus da THB, RMOD-K, esse elemento apresenta melhores resultados para a maioria dos casos práticos. Uma representação desse elemento, com a numeração e a numeração de faces utilizadas pelo *software* MSC Marc é mostrada abaixo.

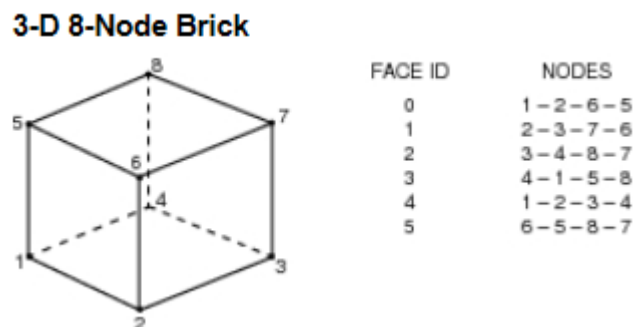


Figura 4 – Numeração dos nós e das faces de um Elemento hexaédrico

Como esse elemento não possui nós intermediários (*midpoints*), o mesmo utiliza funções base trilineares, de uma maneira análoga escreve-se:

$$u_e^h = \sum_{A=1}^{Nne} N^A(x) d_e^A, \quad w = \sum_{B=1}^{Nne} N^B(x) c_e^B \quad (58)$$

Desta maneira o elemento físico Ω^e , o qual provém do domínio físico discretizado é obtido a partir do domínio natural, Ω^ξ através de um mapeamento, como mostra a figura 5.

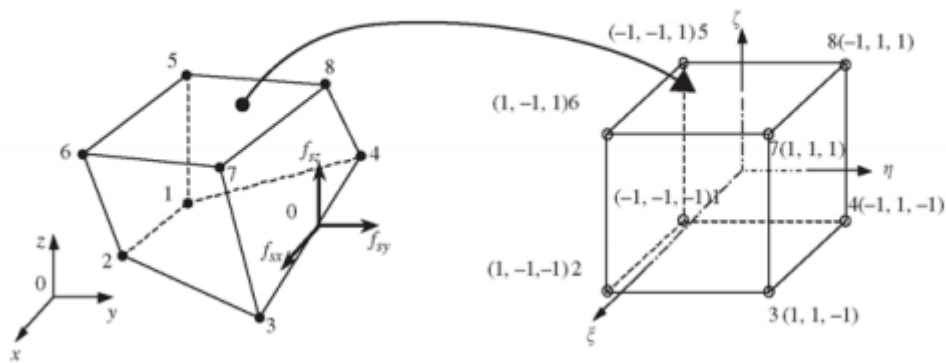


Figura 5 – Mapeamento do domínio natural para o domínio físico

Fonte: MSC Marc Reference Manual (2018).

A funções base podem ser obtidas a partir de formula dos polinômios de Lagrange, equação (15), com um produto tensor. Desta maneira, para esse exemplo 3D as bases terão o seguinte formato:

$$N^1(\xi, \eta, \psi) = \frac{1}{8}(1 - \xi)(1 - \eta)(1 - \psi) \quad (59)$$

Portanto, o mapeando de Ω^ξ para Ω^e é obtido da seguinte maneira:

$$x_e(\xi) = \sum_{A=1}^{Nne} N^A(\xi) x_e^A \quad (60)$$

Desta maneira a forma fraca pode ser escrita através de um somatório de integrais:

$$\sum_e^{Nel} \int_{\Omega^e} w_{,i}^h j_i^h dV = \sum_e^{Nel} \int_{\Omega^e} w^h f dV - \sum_{e \in E_n} \int_{\delta\Omega_j} w^h j_n dS \quad (61)$$

Na qual, Para as condições de contorno do tipo Neumann, $e \in E_n$, se $\delta\Omega^e \cap \delta\Omega_j \neq \emptyset$. Percebe-se também, que de maneira análoga ao problema unidimensional, as derivadas da solução experimental e da função peso devem ser calculadas. Essas estão calculadas a seguir.

$$u_{e,i}^h = \sum_{A=1}^{Nne} N_{,i}^A(\xi) x_e^A \quad (62)$$

$$w_{e,i}^h = \sum_{A=1}^{Nne} N_{,i}^A(\xi) c_e^A \quad (63)$$

Para simplificar a notação utilizada, as seguintes renomeações são feitas:

$$\begin{Bmatrix} \xi \\ \eta \\ \psi \end{Bmatrix} = \begin{Bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{Bmatrix}$$

$$\underline{x} = \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}$$

Utilizando essa nova noção as funções de interpolação podem ser escritas da seguinte maneira:

$$N_{,i}^A = \frac{\delta N^A}{\delta x_i} = \frac{\delta N^A}{\delta \xi_I} \frac{\delta \xi_I}{\delta x_i} \quad (64)$$

Todavia, percebe-se que o termo $\frac{\delta \xi_I}{\delta x_i}$ não pode ser calculado de maneira direta. Para obter o valor do mesmo, deve-se fazer o uso da Matriz Jacobiana. A matriz Jacobiana consiste das derivadas do sistema global de coordenadas em relação ao sistema de coordenadas natural, desta forma, ela é uma matriz de transformação. Neste trabalho, a matriz jacobiana será representada por $\underline{J}$.

É possível encontrar a Matriz Jacobiana através da inversão de sua matriz inversa, a qual é encontrada quando se escreve as funções de forma em notação matricial. Desta forma a Matriz jacobiana é a seguinte:

$$\underline{J} = \begin{bmatrix} \frac{\delta x_1}{\delta \xi_1} & \frac{\delta x_1}{\delta \xi_2} & \frac{\delta x_1}{\delta \xi_3} \\ \frac{\delta x_2}{\delta \xi_1} & \frac{\delta x_2}{\delta \xi_2} & \frac{\delta x_2}{\delta \xi_3} \\ \frac{\delta x_3}{\delta \xi_1} & \frac{\delta x_3}{\delta \xi_2} & \frac{\delta x_3}{\delta \xi_3} \end{bmatrix} \quad (65)$$

E sua inversa é:

$$\underline{J}^{-1} = \begin{bmatrix} \frac{\delta \xi_1}{\delta x_1} & \frac{\delta \xi_1}{\delta x_2} & \frac{\delta \xi_1}{\delta x_3} \\ \frac{\delta \xi_2}{\delta x_1} & \frac{\delta \xi_2}{\delta x_2} & \frac{\delta \xi_2}{\delta x_3} \\ \frac{\delta \xi_3}{\delta x_1} & \frac{\delta \xi_3}{\delta x_2} & \frac{\delta \xi_3}{\delta x_3} \end{bmatrix} \rightarrow \underline{J}_{li}^{-1} = \frac{\delta \xi_l}{\delta x_i} \quad (66)$$

Algumas propriedades importantes da Matriz Jacobiana são:

- O determinante da matriz, $\det(J)$, é o fator de escala. Ele representa o quanto o elemento do domínio natural mudou de volume durante a transformação. Portanto, pode-se assumir:

$$dV = \det(J) dV^\xi \quad (67)$$

- Se o determinante não for constante, a integração numérica utilizando a Quadratura Gaussiana não será exata.
- O *scaled Jacobian* é uma das métricas mais importantes para avaliar a qualidade da malha, pois ela avalia o quanto a matriz de rigidez perde acuracidade durante a integração.

Sabendo da existência da Matriz Jacobiana, assim como sua aplicação, podemos escrever as integrais da Forma Fraca do problema 3D utilizando as derivadas encontradas. Primeiramente, reescreve-se a integral em função das derivadas da solução experimental e da função de interpolação, w :

$$\int_{\Omega^e} w_{,i}^h j_i^h dV = - \int_{\Omega^e} w_{,i}^h K u_{,j}^h dV \quad (68)$$

Aplicando as derivadas, encontradas na equação (64) e a como também a derivada das funções de forma encontradas na equação (62) e (63), obtém-se:

$$\int_{\Omega^e} w_{,i}^h j_i^h dV = - \sum_{A,B} c_e^A \left(\int_{\Omega^e} N_{,I}^A \xi_{I,i} K_{ij} N_{,J}^B \xi_{J,j} dV \right) d_e^B \quad (69)$$

Note que nesta expressão, dentro da integral a convenção do Somatório de Einstein deve ser aplicada nos índices que correspondem tanto as derivadas em relação ao domínio físico, i , quanto ao domínio natural, I . Agora, utilizando das propriedades da Matriz Jacobiana para completar a mudança de variáveis, e a equação (68).

$$\begin{aligned} \int_{\Omega^e} w_{,i}^h j_i^h dV &= - \sum_{A,B} c_e^A \left(\int_{\Omega^\xi} N_{,I}^A \xi_{I,i} K_{ij} N_{,J}^B \xi_{J,j} \det [J(\underline{\xi})] dV^\xi \right) d_e^B = \dots \\ &- \sum_{A,B} c_e^A \left(\int_{\xi_1=-1}^1 \int_{\xi_2=-1}^1 \int_{\xi_3=-1}^1 N_{,I}^A \xi_{I,i} K_{ij} N_{,J}^B \xi_{J,j} \det [J(\underline{\xi})] d\xi_1 d\xi_2 d\xi_3 \right) d_e^B \quad (70) \end{aligned}$$

As integrais irão resultar em um escalar pois, o somatório de Einstein irá eliminar os índices I, i, J, j , todavia A e B não irão desaparecer. Portanto, pode-se escrever a equação (71) como:

$$- \sum_{A,B} c_e^A K_e^{AB} d_e^B = - \underline{c_e^T} \underline{K_e} \underline{d_e} \quad (70)$$

Realizando-se um procedimento análogo para a segunda integral da equação (61), obtém-se:

$$\int_{\Omega^e} w^h f dV = \underline{c_e^T} \underline{F_e^{int}} \quad (71)$$

$$\int_{\delta\Omega_j} w^h j_n dS = \underline{c_e^T} \underline{F_e^j} \quad (72)$$

Onde $\underline{F_e^{int}}$ e $\underline{F_e^j}$ representam as contribuições na matriz de força devido a geração de calor e fluxo de calor, respectivamente. Aplicando os valores das integrais na equação (61):

$$\sum_e^{Nel} \underline{c_e^T} \underline{K_e} \underline{d_e} = - \sum_e^{Nel} \underline{c_e^T} \underline{F_e^{int}} + \sum_{e \in E_n} \underline{c_e^T} \underline{F_e^j} \quad (73)$$

De uma maneira análoga, muda-se o sistema de índices local para global e acoplam-se as matrizes de rigidez. O acoplamento é feito através da adição da contribuição de cada nó do elemento, para o seu respectivo grau de liberdade na matriz de rigidez que representa o problema. Após isso, utiliza-se o argumento que a expressão é válida para $\forall c \in \mathbb{R}^{N_{np}-N_D}$, resultando na equação padrão de elementos finitos:

$$\underline{K} \underline{d} = \underline{F} \quad (74)$$

3 METODOLOGIA

No capítulo anterior foi realizada uma extensa e detalhada revisão bibliográfica para fundamentação teórica em relação ao método de elementos finitos, o qual foi aplicado tanto no desenvolvimento do código quanto na análise dos resultados obtidos. Todo este conteúdo foi essencial para desenvolvimento deste trabalho.

Nesse trabalho, o código foi desenvolvido utilizando o Ambiente de desenvolvimento integrado *Microsoft Visual Studio 2013* com auxílio da ferramenta Git para o versionamento do programa ao longo de seu desenvolvimento. Finalmente, a linguagem escolhida para o desenvolvimento do código foi C++, visto a sua maior velocidade, quantidade de trabalho já realizado na linguagem e compatibilidade com o programa previamente desenvolvido.

Os passos seguidos na metodologia deste trabalho foram os seguintes:

- a) Desenvolver um programa de elementos finitos para resolver problemas termodinâmicos que suporte as condições de contorno: Temperatura, fluxo de calor, geração de calor e convecção.
- b) Utilizando uma malha simplificada, verificar resultados do programa desenvolvido com um software comercial, o qual já tenha passado por testes de validação.
- c) Adaptação do programa para ler a arquivos do programa previamente desenvolvido
 - a. Habilitar importação de malha
 - b. Utilização da mesma numeração para nós e faces
 - c. Utilização dos mesmos índices de posição do elemento no pneu
 - d. Visualizar malha e resultados no RMOD-K
- d) Adaptar a malha obtida para o problema termodinâmico (adição de camadas)
- e) Verificação de resultados com software comercial

3.1 Desenvolver um programa de elementos finitos para problemas termodinâmicos

O desenvolvimento de um Software para a solução de um problema de elementos finitos, em suma é criar um programa que, dado a malha e condições de contorno, retorne o resultado de deslocamento e/ou temperatura nodal para cada nó.

A malha, de uma maneira geral, consiste de duas matrizes. A primeira matriz, a qual pode ser chamada de lista de nós, contém os nós e suas posições no espaço. Visando reduzir memória necessária, os índices da matriz representavam o número do nó, desta maneira a lista de nós terá a dimensão $N_{nós} \times 3$. A outra matriz é chamada de matriz de conectividade, a qual armazena as informações do elemento em relação aos nós. Em outras palavras, esta matriz informa ao programa quais os nós formam a cada elemento. No desenvolvimento do programa supracitado, essa matriz também foi utilizada para armazenar informações acerca de material do elemento e dos índices que indicam sua posição no pneu.

O papel da malha no programa é ditar quais valores serão colocados na matriz de rigidez do problema. Esses valores são calculados a partir da matriz de rigidez de cada elemento, a quais são calculadas pela equação abaixo, obtida anteriormente:

$$\int_{\xi_1=-1}^1 \int_{\xi_2=-1}^1 \int_{\xi_3=-1}^1 N_{,I}^A \xi_{I,i} K_{ij} N_{,J}^B \xi_{J,j} \det \left[\underline{J} \left(\underline{\xi} \right) \right] d\xi_1 d\xi_2 d\xi_3 \quad (75)$$

Essas integrais são calculadas utilizando o método de Quadratura Gaussiana, desta forma alguns valores de ξ serão substituídos nessa equação, e essa será calculada como uma soma da função nos pontos gaussianos. Como os pontos gaussianos são definidos pelo método da quadratura, as funções de interpolação e suas derivadas, representarão um grande custo computacional caso sejam calculadas a cada momento que são chamadas.

Desta forma, visando reduzir a custo computacional do programa, as funções e suas derivadas são previamente calculadas nos pontos Gaussianos utilizando o *software Maxima*. Os resultados das funções de forma são guardados em matrizes, as quais são fornecidas ao programa durante sua inicialização. As matrizes que representam as condições de contorno Neumann, por sua vez, são calculadas de maneira análoga.

De posse das duas matrizes, que são Matriz de Rigidez e Matriz Força, $\underline{K}$ e $\underline{F}$ respectivamente, o programa agora deve aplicar as condições de contorno essenciais. As condições de contorno do tipo Dirichlet (ou essenciais), as quais no caso deste programa tratam-se de temperatura fixa, são aplicadas na matriz dos graus de liberdade. Sabendo da temperatura dos nós com essa condição de contorno, pode-se resolver parte do sistema linear. Desta forma alguns índices da matriz de rigidez são multiplicados pelas temperaturas de seus respectivos nós e passam para o lado direito da equação. Agora, a matriz de rigidez que antes era singular, possui um determinante diferente de zero. Além disso, a mesma pode ser reescrita na forma

reduzida, a qual considera somente os graus de liberdade sem temperatura fixa. Consequentemente, a matriz Força deve conter somente os índices que correspondem aos graus de liberdade livres.

Finalmente, tendo em mão todas as matrizes necessárias, foi utilizando um algoritmo, retirado de uma biblioteca *open source*, baseado no Método do Gradiente Conjugado. Esse algoritmo resolve o sistema linear, retornando a matriz das temperaturas que ainda não estavam fixadas. De posse das temperaturas que antes eram desconhecidas e das que foram fixadas, tem-se a temperatura em cada nó do sistema.

3.2 Verificando resultados com *software* comercial

Visando verificar os cálculos realizados pelo código, vários casos teste foram realizados utilizando um *software* comercial. Utilizando o outro *software*, o mesmo problema era especificado, e o resultado de ambos os softwares eram comparados. O programa escolhido para a realização da comparação foi o MSC Marc. Esse *software* foi escolhido pois, além de ser reconhecido por resolver problemas não-linear com grandes deformações, possui uma interface que permite que o problema seja especificando a mesma malha. Desta forma, a malha foi especificada utilizando a mesma matriz com as posições nodais e a matriz de conectividade. MSC Marc, ainda disponibiliza uma ferramenta que exporta o resultado em cada nó da malha.

O primeiro teste foi realizado em uma estrutura trapezoidal simples, mostrada na figura abaixo. O modelo tem 100 milímetros na direção X, 50 milímetros na direção Z. Na direção Y, o modelo a maior aresta possui 150 milímetros, enquanto a menor possui 100 milímetros. A malha contém 100 nós e 48 elementos hexaédricos. Cada elemento tem uma condutividade isotrópica constante K .

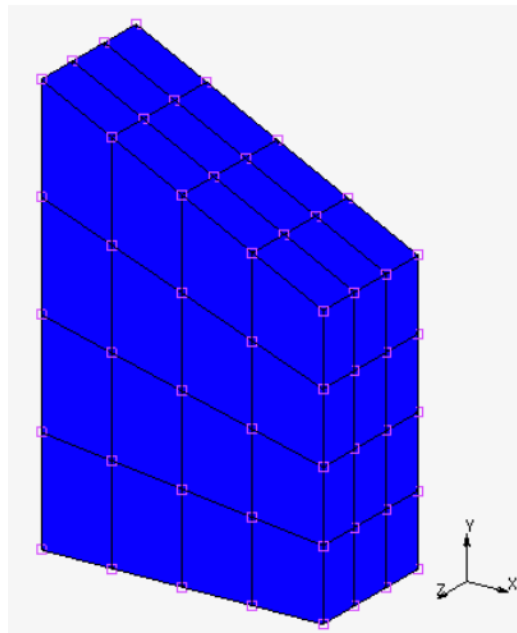


Figura 6 – Malha da estrutura trapezoidal utilizada

Fonte: Própria do autor (2018).

O segundo caso teste, consiste em um Toróide, o qual possui um formato mais similar com um pneu. O toróide possui um diâmetro de 14 mm com seu centro distante 15mm o centro de rotação. A malha do modelo contém 7140 nós e 6400 elementos hexaédricos. A malha foi gerada pelo MSC Marc, utilizando partições a cada 20 graus.

Para ambos os casos foram testadas as condições de contorno:

- Temperatura fixa
- Convecção
- Fluxo de calor
- Geração de calor

3.3 Adaptação do programa para ler as malhas do programa previamente desenvolvido

Visto que a intenção desse código é criar um programa de elementos finitos que alimente o programa já desenvolvido para cálculos mecânicos estruturais com temperaturas nodais, a malha utilizada nesse software deve ser intercambiável. Desta forma, o programa deve ser capaz

de ler a malha fornecida pela parte mecânica estrutural. A primeira linha deste arquivo tem informações acerca do número de nós e de elementos da malha. Com estas informações, é possível limitar os laços dentro do programa e ler a malha.

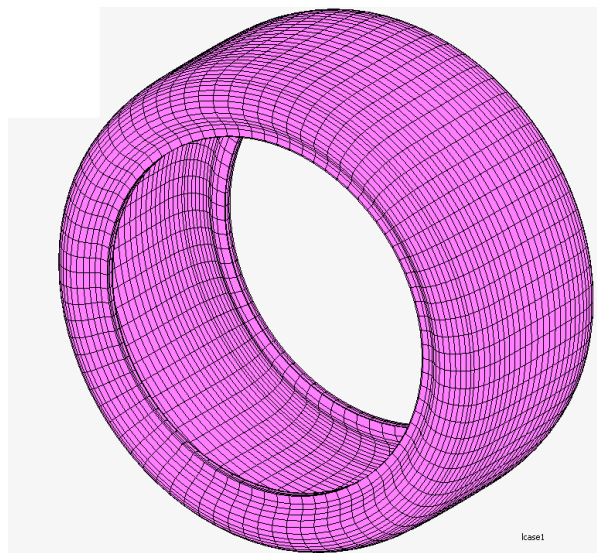


Figura 7 – Malha original do pneu

Fonte: Própria do autor (2018).

Uma atenção especial deve ser dada a matriz de conectividade, pois essa informará a os nós de cada elemento, como também um índice que indica em que posição cada elemento está na estrutura do pneu. Essa posição, irá permitir no futuro especificar quais elementos na malha deseja-se refinar. Tomando esses cuidados, é possível adaptar o programa para que o mesmo leia a malha, a qual está apresentada na figura acima.

3.4 Adaptar a malha obtida para o problema termodinâmico (adição de camadas)

Sabendo que o pneu é composto de várias camadas compostas de fibras metálicas, é necessário a inserção de camadas finas na malha. Visto que essas camadas tem uma maior densidade de material metálico, espera-se que ela tenha uma maior condutividade, tendo uma influência considerável na acurácia do programa.

Para fazer tal procedimento, foi desenvolvida uma classe chamada de *Tmesh*, a qual significa Malha Termodinâmica. Essa classe possui a malha modificada para o problema

termodinâmico, como também as funções usadas na manipulação dos elementos da malha. Como variáveis essa classe contém o número de nós, número de elementos e número de elementos na secção transversal do pneu. Essa classe também possui funções que lidam com a alocação das matrizes, o que é muito importante quando se lida com refinamento de malhas.

Finalmente, a função mais importante, *addLayer* (Adicionar Camada), é a função responsável por adicionar a camada de material metálico à malha. Essa função tem como entrada o elemento especificado, a espessura da camada que desejasse inserir e a condutividade do material inserido. Desta maneira, a função faz um *loop* sobre as arestas do elemento especificado, o *loop* irá percorrer as arestas na seguinte sequência: 1 – 4, 2 – 3, 5 – 8, 6 – 7. O local que os nós deverão ser inseridos são calculados a partir da posição nodal dos nós da aresta. Os novos nós serão inseridos no final da lista de nós, desta maneira a malha modificada terá os mesmos nós que a original, e as modificações serão nós extras, os quais não prejudicam a intercambiabilidade de informações nodais entre as malhas.

Utilizando os novos nós, o elemento especificado é dividido em 3, como ilustrado na figura abaixo. Para que os novos elementos sejam representados corretamente, a matriz de conectividade deve ser modificada para o antigo elemento, e dois elementos deverão ser adicionados na matriz de conectividade. Portanto, o número de nós na malha aumenta em 8 e o número de elementos em 2, para cada camada adicionada em um elemento. O procedimento é ilustrado abaixo na Figura (8). A Figura (9), por sua vez, ilustra a adição de uma camada de 0.1mm de espessura, em um conjunto de 4 elementos.

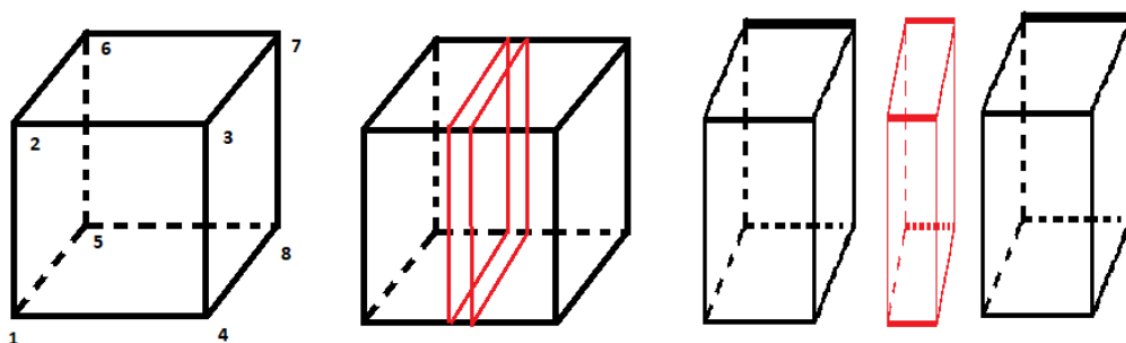


Figura 8 – Procedimento de adição de camadas

Fonte: Própria do autor (2018).

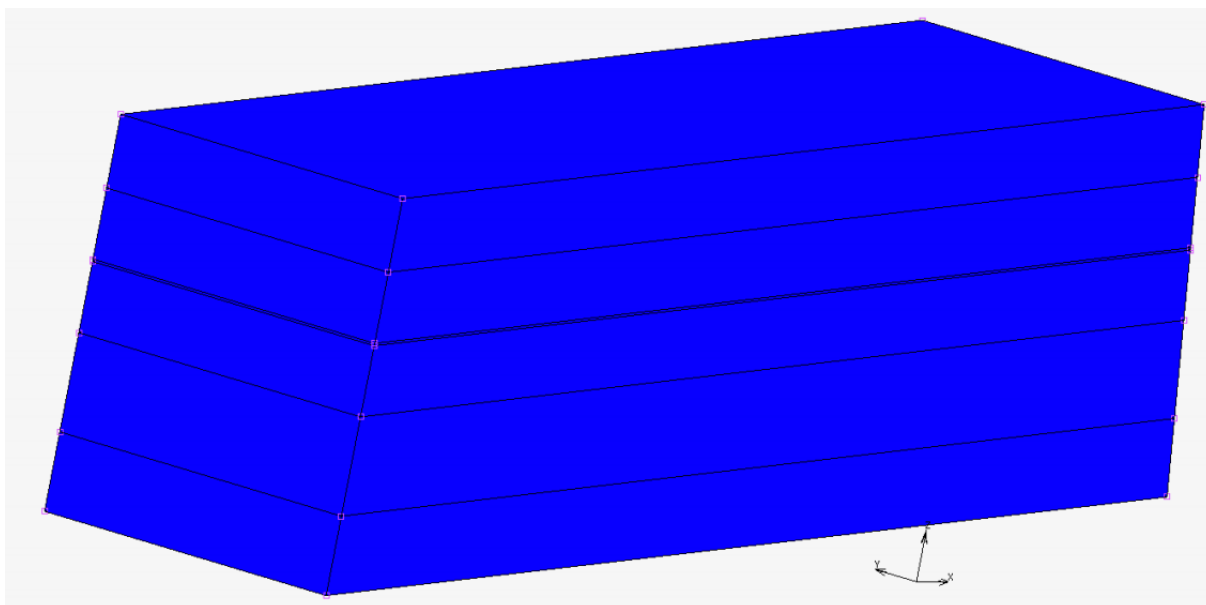


Figura 9 – Malha com camada de 0.1mm de espessura adicionada

Fonte: Própria do autor (2018).

Todavia, esse procedimento irá produzir nós duplicados em elementos adjacentes, pois a operação de refino é feita de elemento em elemento, não existindo comunicação entre elementos adjacentes. Desta forma, foi criada uma função na classe *Tmesh* que remove os nós duplicados, testando a distância entre os mesmos, e após a remoção é feita uma renumeração na malha. Visto que o processo de criação da malha é feito uma única vez para cada caso teste, essa ineficiência computacional não causará problemas no tempo computacional do programa.

3.5 Verificação de resultados com software comercial

Finalmente, para verificar a malha modificada, novos testes foram realizados utilizando o MSC Marc. Esses testes foram análogos aos explicados na sessão 3.2, todavia a malha teve uma atenção maior nesses testes. Foi verificado se existiam elementos avessados (*inside out*), se a manipulação da malha foi como esperado e se os nós estavam com a numeração correta.

4 RESULTADOS E DISCUSSÃO

Como já citado na seção anterior, os primeiros testes utilizaram malhas simples com o intuito de verificar os resultados produzidos pelo código desenvolvido. Os primeiros testes foram aplicados em uma estrutura trapezoidal, mostrada na Figura 6. Ao total foram realizados 13 testes na estrutura trapezoidal, os quais testaram desde os cálculos das matrizes Jacobianas para as faces e os volumes até temperaturas nodais. Os 10 primeiros testes foram realizados utilizando apenas condições de contorno de fluxo de calor e temperatura fixa. Desta forma, através desses foi possível verificar que:

- As condições de contorno Essenciais estavam sendo aplicadas corretamente;
- Os cálculos de área e volume estavam corretos;
- Matriz de força para condições de Fluxo de calor corretamente calculada;
- Matriz com temperaturas nodais corretamente arranjada;

Ao final de cada teste, todas as temperaturas nodais calculadas entre os resultados de ambos os programas são comparadas através de uma tabela *excel*. A métrica de comparação usada para as temperaturas nodais foi o erro percentual, considerando os valores obtidos no software comercial como valores corretos. A figura abaixo mostra as faces selecionadas e os resultados de temperatura nodal em um dos 10 testes realizados para a avaliação do resultado para fluxo de calor. A maior diferença encontrada foi de 0,00410%, como pode ser conferido na Tabela 1 abaixo. Como essa diferença em valores absolutos representa menos de 1 μK , pode-se considerar que o programa teve resultados compatíveis com o *software* comercial.

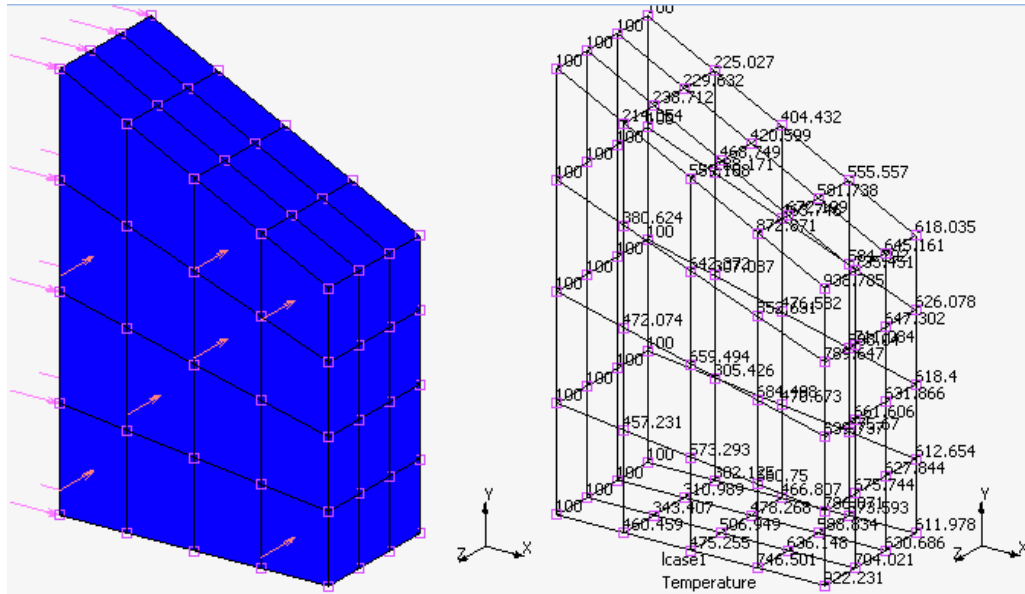


Figura 10 – Malha da geometria trapezoidal com faces selecionadas com fluxo de calor (esquerda) e resultados (direita)

Fonte: Própria do autor (2018).

A seguir, foram realizados dois testes de convecção e um teste de geração de calor. Visto que, os cálculos das matrizes jacobianas já haviam sido verificados com os testes utilizando fluxo de calor, os testes de convecção só foram realizados para verificar a matriz de rigidez e a matriz força gerada pela convecção. O teste realizado para geração de calor por sua vez, verificou mais uma vez que a matriz jacobiana usada para calcular volumes e a matriz de força gerada pela geração de calor. Os dois testes de convecção apresentaram como maiores diferenças as seguintes porcentagens: 0,00187% e 0,00084%. Já o teste de geração de calor, figura 12, apresentou uma diferença nodal máxima na temperatura de 0,0047%, como também pode ser visualizado na Tabela 1. Sabendo que essas diferenças se encontram na ordem de μK , sabe-se que ambas as funções foram implementadas com sucesso.

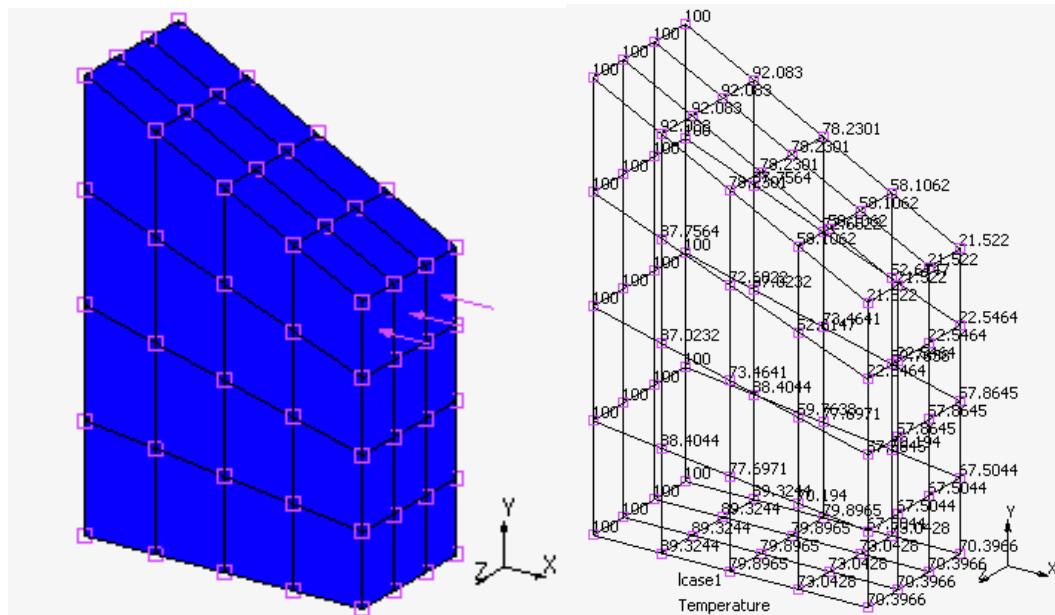


Figura 11 – Malha da geometria trapezoidal com faces selecionadas com convecção (esquerda) e resultados (direita)

Fonte: Própria do autor (2018).

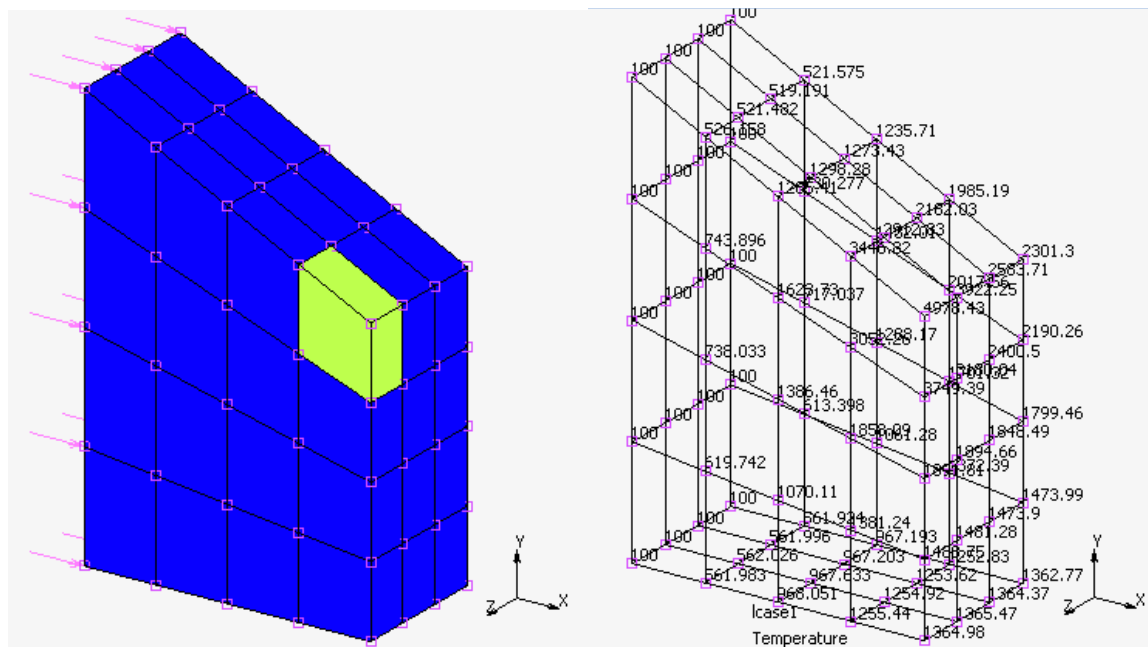


Figura 12 – Malha da geometria trapezoidal com elemento selecionado como gerador de calor (esquerda) e resultados (direita)

Fonte: Própria do autor (2018).

Após isso, foram realizados os testes no Toróide. Neste, foram realizados dois testes, o primeiro utilizou somente temperatura fixa e fluxo de calor como condições de contorno. Já o segundo teste utilizando o toróide, contém todas as condições de contorno implementadas (Geração de calor, Fluxo de calor, convecção e temperatura fixa). As maiores diferenças percentuais foram 0,00237% e 0,00433% para as respectivas simulações, como pode ser visto na Tabela 1. Desta maneira, com esses dois testes no toróide, foi possível verificar os calculos para uma malha mais complexa e também o resultado para mais de uma condição de contorno aplicada.

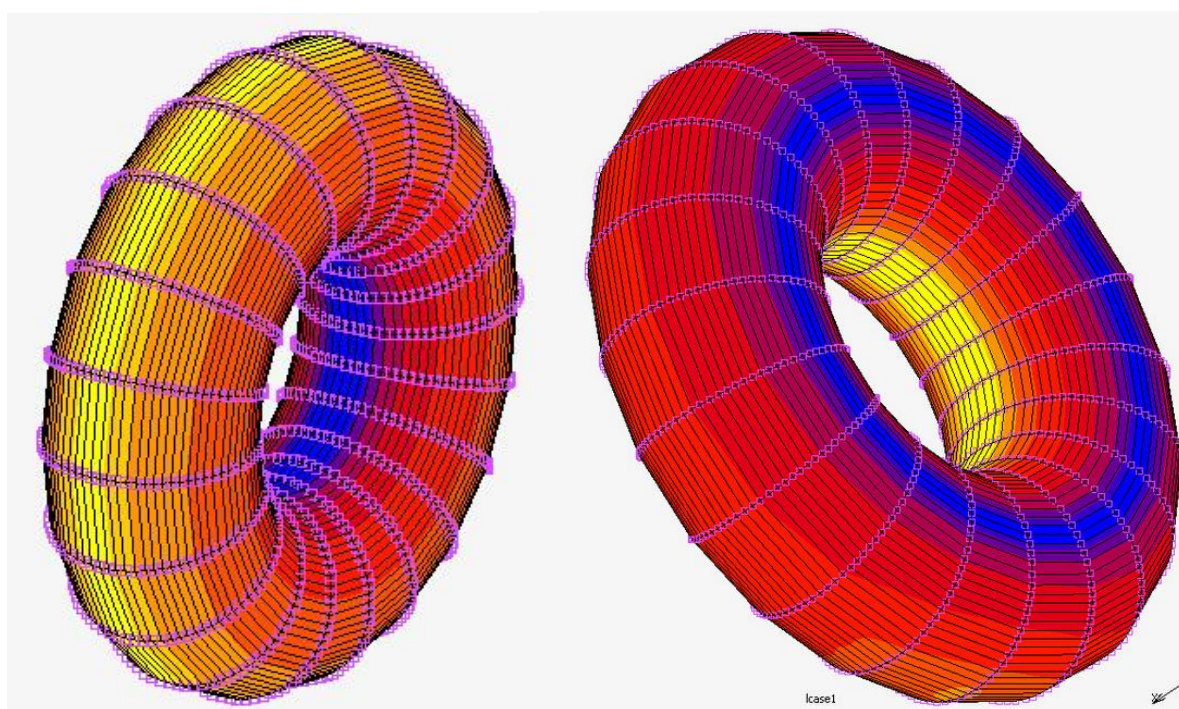


Figura 13 – Resultado qualitativo da simulação no toróide, somente fluxo de calor (esquerda) e todas condições de contorno disponíveis (direita)

Fonte: Própria do autor (2018).

Finalmente, foram realizados os testes na malha utilizada pelo solver mecânico do *software* RMOD-K. O primeiro teste foi realizado na malha sem a adição de camadas entre os elementos, sendo assim a malha original. O teste foi realizado com condições de contorno semelhantes a condições de operação do pneu em regime permanente. As condições de contorno, as quais também foram utilizadas foram as seguintes:

1. Temperatura fixa nos nós em contato com a roda

2. Convecção nos elementos da banda externa, com temperatura ambiente de 30 e um coeficiente de convectividade de 50
3. Convecção nos elementos da banda interna, com temperatura ambiente de 50 e um coeficiente de convectividade de 30

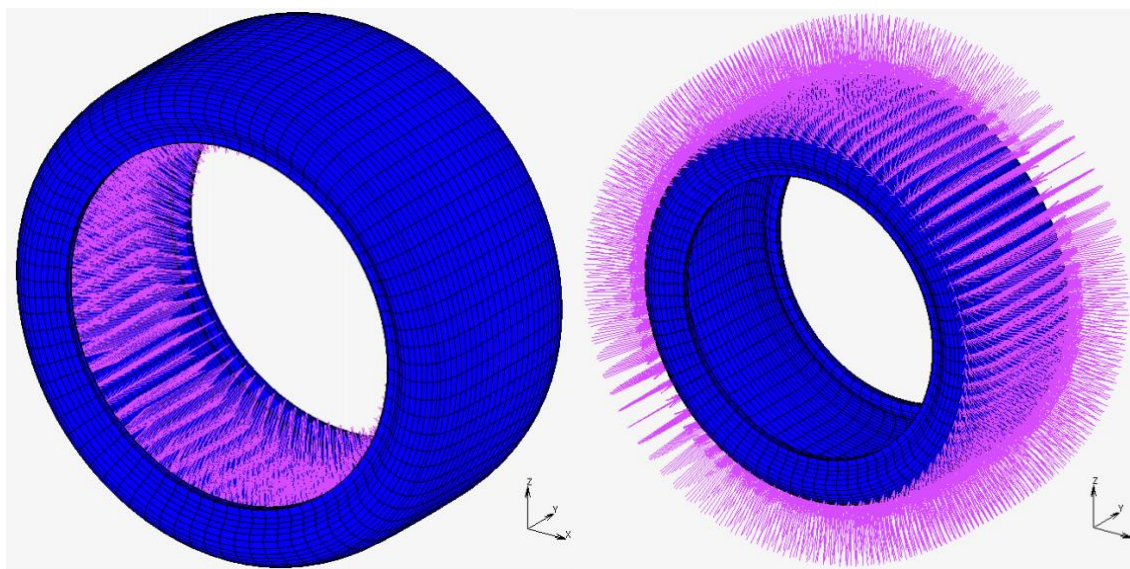


Figura 14 – Condições de contorno utilizadas na simulação das malhas de pneu

Fonte: Própria do autor (2018).

De maneira análoga, foram comparadas todas as temperaturas nodais obtidas pelo código e pelo *software* MSC Marc. Como também pode ser visualizado na Tabela 1, a maior diferença percentual foi de 0,00433%, o que mostra que os resultados obtidos pelo código desenvolvido coincidem com o *software* comercial. O perfil de temperatura obtido pode ser visualizado na figura abaixo.

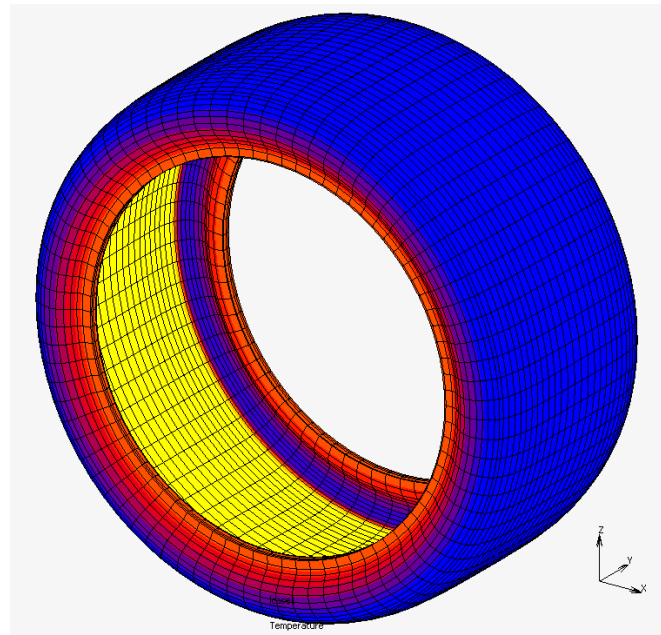


Figura 15 – Perfil de temperatura da simulação na malha original do pneu

Fonte: Própria do autor (2018).

Utilizando as mesmas condições de contorno da simulação na malha sem camadas intermediárias, foi realizado um teste na malha modificada, a qual contém camadas com elementos intermediários. A maior diferença percentual entre as temperaturas obtidas entre o MSC Marc e o código desenvolvido 0,0013% o que resulta em uma diferença absoluta de de 0,00065 Kelvin ($650\mu\text{K}$). As figuras abaixo mostram o perfil de temperatura obtido em uma seção do pneu (Figura 16), e a forma com que o fluxo de calor se propaga ao longo dos elementos com camadas intermediárias (Figura 17).

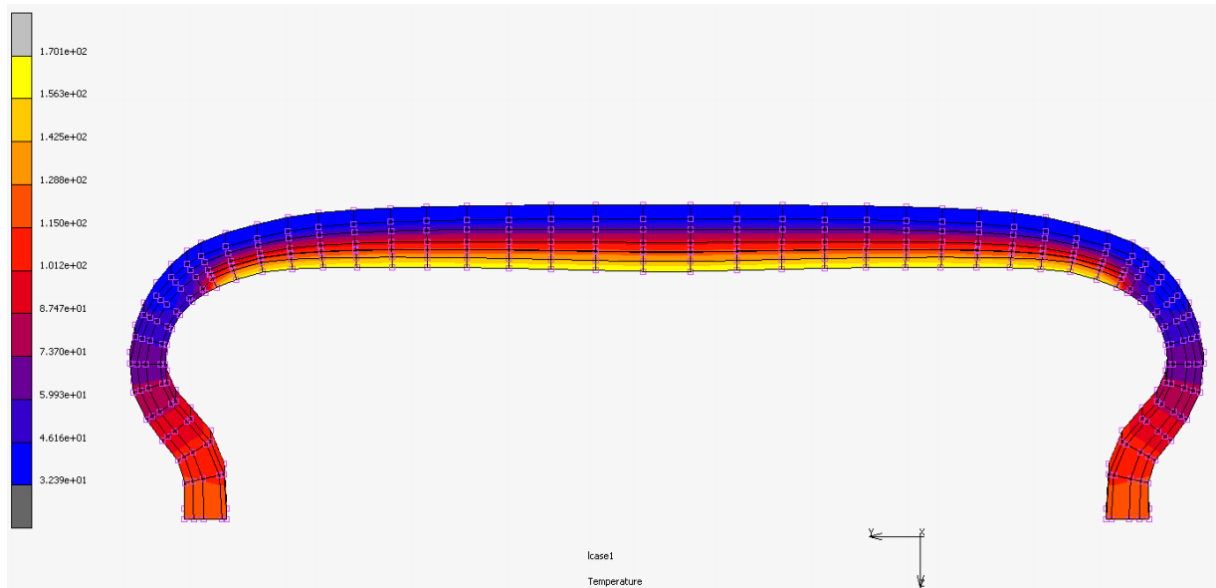


Figura 16 – Resultado de temperaturas nodais na secção transversal do pneu

Fonte: Própria do autor (2018).

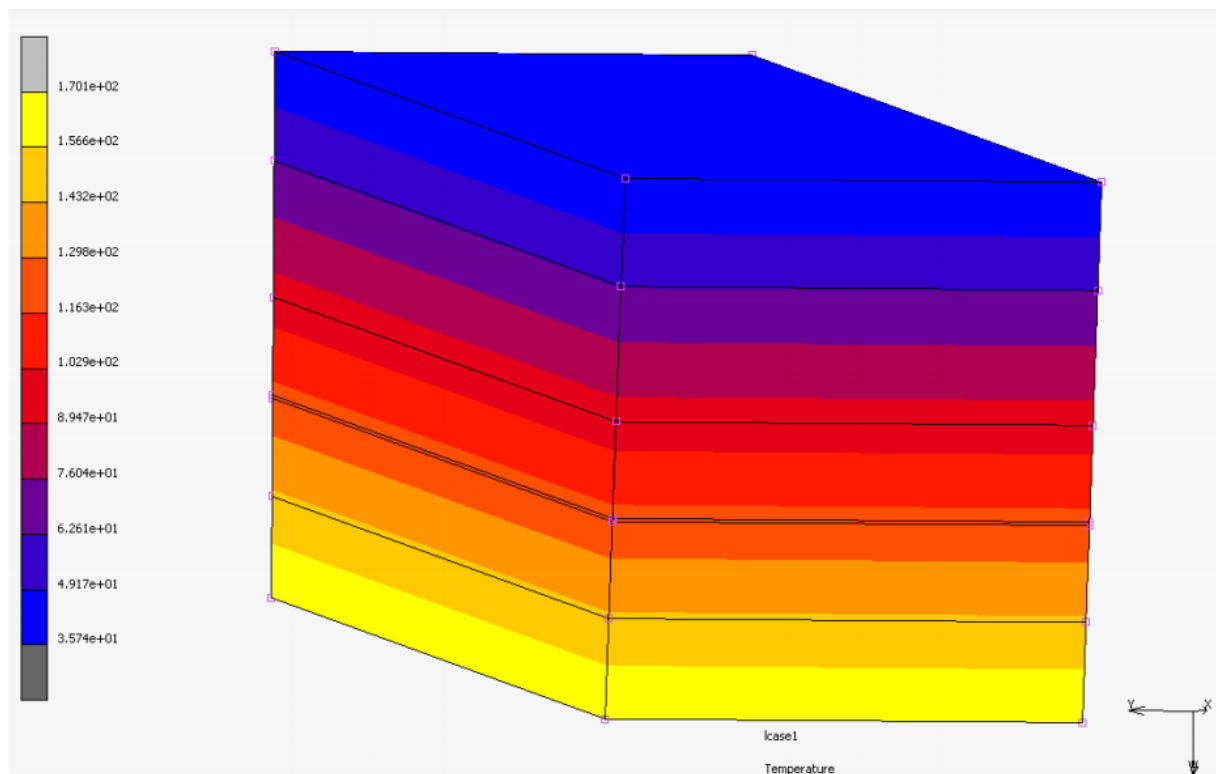


Figura 17 – Resultado de temperatura nodal na malha do pneu em apenas um conjunto de elementos adjacentes

Fonte: Própria do autor (2018).

Finalmente, todos os resultados dos testes realizados podem ser encontrados na tabela abaixo. Sendo os 13 primeiros testes relacionados a geometria trapezoidal, seguido de dois testes no toróide e mais dois testes nas malhas que representam o pneu em si.

Como os valores percentuais irão resultar em uma diferença absoluta na ordem de μK , pode-se considerar que o código desenvolvido foi feliz com os resultados nos casos teste. Além do que, visto que as operações na malha resultaram em uma malha razoável, o código desenvolvido está apto a ser implementado no software já desenvolvido. Finalmente, levando em conta a eficiência do programa em obter os resultados, intercambiabilidade entre as malhas e os refinamentos específicos que o mesmo oferece, pode-se afirmar que o uso do código desenvolvido é mais vantajoso que a implementação de um software comercial para a solução do problema termodinâmico.

Tabela 1 – Diferenças percentuais máximas nos testes realizados.

CASO TESTE		DIFERENÇA MÁXIMA %	TEMPERATURA NO NÓ	
			NO MSC MARC	NO CÓDIGO
Geometria Trapezoidal	Fluxo de calor 1	0,0000348%	239,515533	239,515545
	Fluxo de calor 2	0,0000275%	143,925430	143,925467
	Fluxo de calor 3	0,0000174%	346,467956	346,467931
	Fluxo de calor 4	0,0000276%	157,614028	157,614036
	Fluxo de calor 5	0,0000214%	243,999176	243,999154
	Fluxo de calor 6	0,0000420%	182,380813	182,380845
	Fluxo de calor 7	0,0000363%	175,627731	175,627717
	Fluxo de calor 8	0,0000242%	922,230529	922,230489
	Fluxo de calor 9	0,0000261%	163,420364	163,420333
	Fluxo de calor 10	0,0000410%	180,319351	180,319324
	Convecção 1	0,0000187%	23,310695	23,310655
	Convecção 2	0,0000084%	73,042831	73,042881
	Geração de calor	0,0000470%	1985,192382	1985,192410
Toróide 1		0,0000237%	247,286102	247,286122
Toróide 2		0,0000472%	229,108871	229,108898
Pneu sem camada		0,0000433%	116,747047	116,747092
Pneu com camada		0,0012753%	51,068693	51,068706

Fonte: Produzida pelo autor (2018).

5 CONCLUSÃO

O estudo proposto permitiu o desenvolvimento e validação de um código utilizado para resolver problemas termodinâmicos utilizando o MEF. Relembrando objetivo deste código, o qual funcionará como uma extensão para um programa já desenvolvido, é aumentar a acuraria do RMOD-K através da implementação da abordagem multifísica no modelo.

Diante do fato que o programa foi capaz de ler a malha e adaptá-la para a aplicação desejada, mantendo a numeração de seus nós compartilhados, têm-se as temperaturas de cada nó na malha do programa mecânico. Portanto, utilizando funções de interpolação e técnicas de integração é possível avaliar a temperatura média em um elemento e ajustar as propriedades mecânicas do modelo para tal.

Por sua vez, a validação do programa, a qual contou com auxílio de um *software* comercial para comparação dos resultados, apresentou excelentes resultados. Como é possível verificar na Tabela 1 a maior diferença percentual encontrada foi de 0,0013%, a qual representa uma diferença absoluta nos resultados de 650 μK . Desta maneira, pode-se aferir que os resultados obtidos pelo programa são confiáveis e poderão ser usados na abordagem multifísica.

Finalmente, através deste trabalho foi possível cumprir os objetivos especificados no capítulo 2, bem como também serve como fonte de inspiração e impulso para próximos trabalhos continuando este tema. As sugestões para trabalhos futuros se encontram no capítulo seguinte.

6 SUGESTÕES PARA TRABALHOS FUTUROS

O comportamento do pneu muitas vezes é um problema que depende do tempo, como a maioria dos problemas do mundo real. Portanto, para representar de maneira precisa as propriedades do material com relação a sua temperatura, uma abordagem que depende do tempo deve ser utilizada. Por exemplo, a maior condução de calor nas camadas metálicas, é melhor visualizada em uma abordagem que depende do tempo.

É possível visualizar esse problema, aplicando uma condição de fluxo de calor somente na metade do pneu, como mostrado na figura 18. Desta forma, espera-se que o calor seja conduzido mais rapidamente nas camadas de com maior densidade metálica (ilustrada na figura 19 e 20). O resultado esperado é mostrado na figura 21.

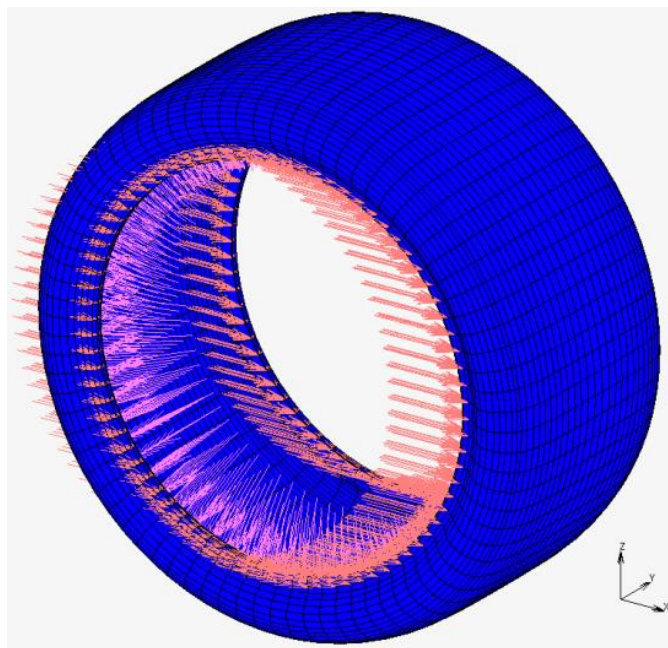


Figura 18 – Condições de contorno para o teste transiente

Fonte: Própria do autor (2018).

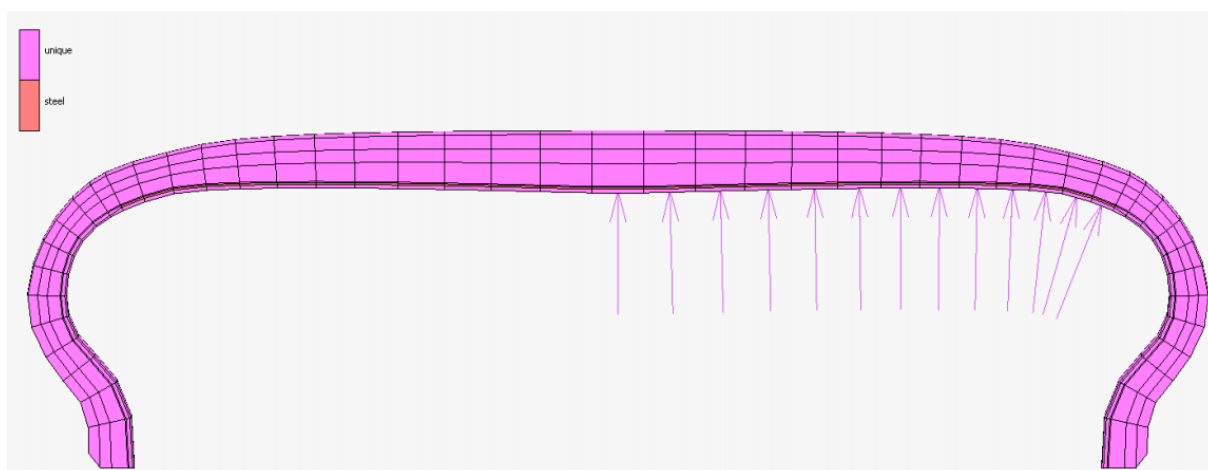


Figura 19 – Corte transversal da malha com condições de contorno

Fonte: Própria do autor (2018).

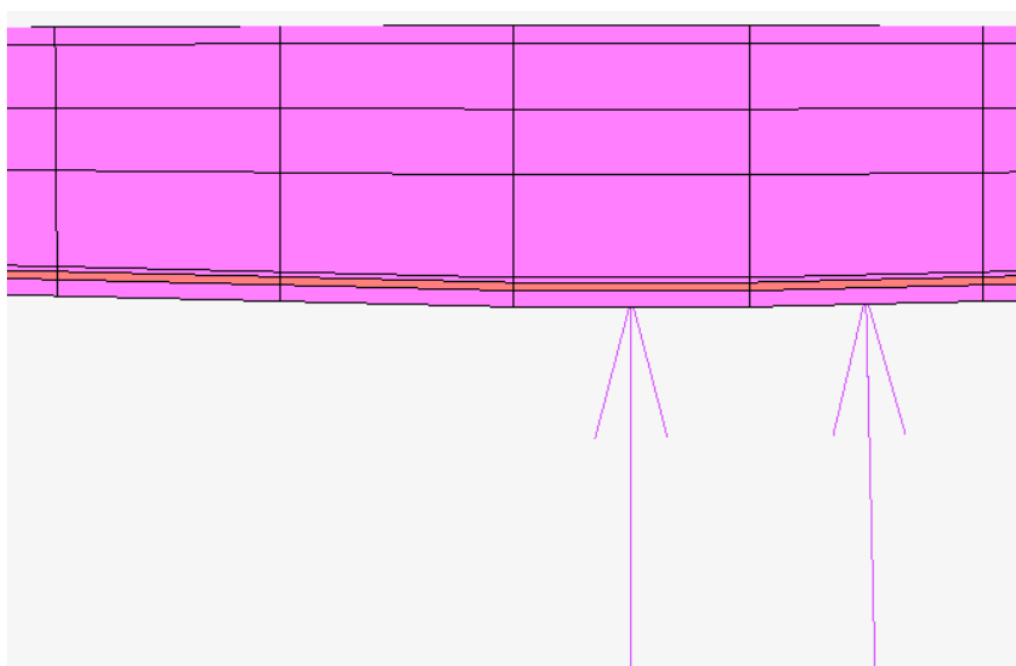


Figura 20 – Corte transversal com zoom para visualização da camada metálica

Fonte: Própria do autor (2018).

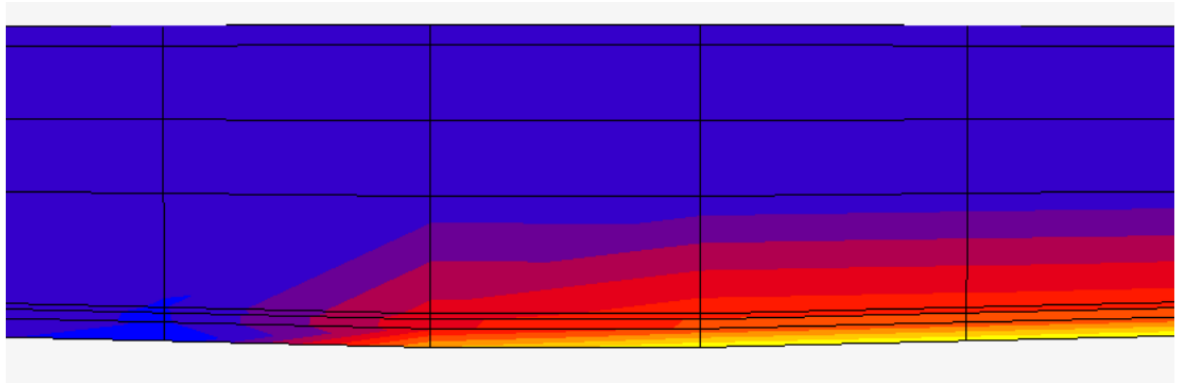


Figura 21 – Perfil de temperatura esperado para simulação transiente especificada

Fonte: Própria do autor (2018).

Portanto, possíveis continuidades para o trabalho são:

1. Comparação do desempenho entre os diferentes esquemas de discretização do tempo implícito e explícito
2. Implementação de uma solução dependente do tempo utilizando o solver com o algoritmo de discretização escolhido
3. Redução do tempo de processamento e memória necessária através de novas técnicas de alocação de matrizes
4. Desenvolvimento de interface capaz de facilitar as operações na malha

REFERÊNCIAS

ALMEIDA, Heitor Augusto Carneiro de. **Método dos Elementos Finitos**: Curso online da Universidade do Michigan. 01 sep. 2017, 01 dec. 2017. 56 p. Notas de Aula.

FISH, J.; BELYTSCHKO, T.; **A First Course in Finite Elements**, 1ed. Wiley, 2007.

Flexible Belt (FB). **Flexible Belt (FB)**. Disponível em: < <https://www.rmod-k.com/flexible-belt-fb>>. Acesso: 22.out.2018.

KIM, N.; SANKAR, B.V.; **Introdução à análise e ao projeto em elementos finitos**, 1ed. LTC, 2009

LARSON, M. G.; BENGTZON, F.; **The Finite element Method: Theory, Implementation, and Applications**, 1ed. SPRINGER, 2013.

Measurements. **Measurements**. Disponível em: < <https://www.rmod-k.com/measurements>>. Acesso: 22.out.2018.

Oertel, C.; Frandre, A; “Reifenmodellsystem RMOD-K - Ein Beitrag zum virtuellen Fahrzeug”. **ATZ worldwide**. Vol. 103, n. 11, p. 23-25, 2001.

Oertel, C.; Wei, Y; “Tyre rolling kinematics and prediction of tyre forces and moments: part I –theory and method”. **Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility**. Vol. 50, n. 11, p. 1673-1687, 2012.

Oertel, C.; Wei, Y; “Tyre rolling kinematics and prediction of tyre forces and moments: part II –Simulation and Experiment”. **Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility**. Vol. 50, n. 11, p. 1689-1706, 2012.

Patel et al.; “A new multibody system approach for tire modeling using ANCF finite elements”. **Proceedings of the Institution of Mechanical Engineers Part K Journal of Multi-body Dynamics**. Vol. 230, n. 1, p. 69-84, 2016.

PAVLOU, D. G.; **Essentials of the Finite Element Method for Mechanical and Structural Engineers**, 1ed. ELSEVIER, 2015.

SEDRA, M., “FEM and thermodynamics”. **Trabalho de conclusão de curso, Universidade Técnica de Brandenburgo**, Brandenburg an der Havel, Alemanha, 2016.