

**UNIVERSIDADE DO VALE DO ITAJAÍ
ESCOLA DO MAR, CIÊNCIA E TECNOLOGIA
CURSO DE ENGENHARIA MECÂNICA**

JULIO PATRON WITWYTZKYJ

**MODELAGEM BIFOCAL APLICADA À ENGENHARIA MECÂNICA:
Desenvolvimento de um modelo computacional de condução térmica para uso
educacional no ensino superior**

**Itajaí
2018**

JULIO PATRON WITWYTZKYJ

**MODELAGEM BIFOCAL APLICADA À ENGENHARIA MECÂNICA:
Desenvolvimento de um modelo computacional de condução térmica para uso
educacional no ensino superior**

Trabalho de Iniciação Científica e Tecnológica
apresentado para a obtenção do título de Engenheiro
Mecânico pelo Curso Superior de Engenharia Mecânica
da Escola do Mar, Ciência e Tecnologia da Universidade
do Vale do Itajaí.

Orientador: Adhemar Maria do Valle Filho, Dr.

Itajaí

2018



UNIVALI

UNIVERSIDADE DO VALE DO ITAJAÍ

FOLHA DE APROVAÇÃO

TRABALHO DE INICIAÇÃO CIENTÍFICA E TECNOLÓGICA

**UNIVERSIDADE DO VALE DO ITAJAÍ
ESCOLA DO MAR, CIÊNCIA E TECNOLOGIA
CURSO DE ENGENHARIA MECÂNICA**

JULIO PATRON WITWYTZYJ

**MODELAGEM BIFOCAL APLICADA À ENGENHARIA MECÂNICA:
Desenvolvimento de um modelo computacional de condução térmica para uso educacional no
ensino superior**

**Trabalho de Iniciação Científica e Tecnológica apresentado ao Curso de
Engenharia Mecânica, para a obtenção do grau de Engenheiro Mecânico.**

Aprovado na data da defesa.

**Prof. Adhemar Maria do Valle Filho, Dr.
EMCT**

**Prof. Júlio César Leão, Me.
EMCT**

**Prof. Rudimar Luís Scaranto Dazzi, Dr.
EMCT**

Itajaí, 10 de dezembro de 2018.

ESCOLA DO MAR, CIÊNCIA E TECNOLOGIA

Rua Uruguai, 458 – Centro – Itajaí – SC – CEP 88302-202 – Caixa Postal 360 – Fone: (47) 3341 7541 – Fax: (47) 3341 7715
www.univali.br

*Dedico este trabalho aos meus pais,
as luzes desse mundo,
Ivone e Jon.*

AGRADECIMENTOS

Agradeço ao meu orientador Adhemar Maria do Valle Filho, pela disponibilidade, pelo incentivo, pela contribuição e pelos ensinamentos que tornaram possível a realização deste trabalho, assim como por sua paciência e compreensão ao assumir a orientação deste projeto. Seu entusiasmo e auxílio em relação aos projetos desenvolvidos certamente inspiram a criatividade e o avanço do conhecimento científico.

Agradeço ao professor André Luiz Maciel Santana por ter me apresentado às ferramentas e aos métodos aqui descritos, assim como por todo o auxílio fornecido durante a primeira etapa deste trabalho. Desejo-lhe sucesso em sua busca contínua pela melhoria da educação e pela disseminação do conhecimento.

Agradeço aos demais professores do curso de Engenharia Mecânica da Universidade do Vale do Itajaí, por me beneficiarem com uma pequena parte de todo o seu conhecimento durante a graduação.

Agradeço imensamente aos meus pais, Ivone e Jon, por todo o conhecimento, pela experiência e pela sabedoria transmitidos, assim como por todo o carinho e pelo cuidado dedicado a mim. Agradeço também à minha irmã e aos meus tios, pelo carinho e pelo suporte nos momentos de necessidade.

Agradeço à Priscilla, minha namorada, pelo imenso auxílio, compreensão, carinho e suporte emocional. Empristo a famosa frase de Carl Sagan para dizer que “diante da vastidão do tempo e da imensidão do universo, é um imenso prazer para mim dividir um planeta e uma época com você.”

RESUMO

WITWYTZYK, Julio Patron. Modelagem Bifocal Aplicada à Engenharia Mecânica: Desenvolvimento de um modelo computacional de condução térmica para uso educacional no ensino superior. 2018. 125 f. Trabalho de Iniciação Científica e Tecnológica (Graduação em Engenharia Mecânica) – Escola do Mar, Ciência e Tecnologia, Universidade do Vale do Itajaí, Itajaí, 2018.

Alunos do curso de graduação em Engenharia Mecânica frequentemente apresentam dificuldade em relação ao entendimento do comportamento de fenômenos físicos. Em contrapartida, é sabido que a interação dos estudantes com as simulações computacionais de fenômenos físicos é capaz de proporcionar benefícios ao entendimento do fenômeno estudado. A técnica de Modelagem Bifocal integra experimentação no mundo físico e em ambiente virtual, de forma que o estudante realize um experimento físico, coletando dados através de sensores, ao mesmo tempo em que desenvolve um modelo computacional do mesmo processo. Assim, os dados coletados no experimento físico e na simulação do modelo computacional podem ser comparados em busca da obtenção de um modelo digital que represente o comportamento do fenômeno físico. Neste trabalho um modelo computacional de condução térmica foi criado no *software* NetLogo em conjunto com um experimento físico análogo. Cinco pontos de temperatura foram coletados ao longo de quatro medidas com faixas de temperatura distintas. Ao fim, os valores de temperatura comparados entre o modelo computacional e o experimento físico obtiveram convergência estatística em quatro dos cinco pontos de análise quando avaliados de maneira pareada, sendo necessário que sejam revisados em busca de pequenas correções. Observou-se durante o desenvolvimento que a Modelagem Bifocal capacita o pesquisador a pensar criticamente sobre seu modelo, auxiliando na formulação de hipóteses para as possíveis divergências.

Palavras-chave: Modelagem Bifocal. Simulação Computacional. Experimentação. Aprendizado.

ABSTRACT

Students of the undergraduate course in Mechanical Engineering often find difficulties in understanding the behavior of physical phenomena. In contrast, it is known that the interaction of students with computational simulations of physical phenomena can provide benefits to the understanding of the phenomenon studied. The Bifocal Modeling technique integrates experimentation in the physical world and virtual environment, so that the student performs a physical experiment, collecting data through sensors, while developing a computational model of the same process. Thus, the data collected in the physical experiment and in the simulation of the computational model can be compared to obtain a digital model that represents the behavior of the physical phenomenon. In this study a computer model of thermal conduction was created in NetLogo software in conjunction with an analogous physical experiment. Five temperature points were collected over four measurements with different temperature ranges. At the end, the temperature values compared between the computational model and the physical experiment reached statistical convergence in four of the five points of analysis when evaluated in pairs, with only small corrections being necessary. It was observed during the development that Bifocal Modeling enables the researcher to think critically about its model, helping in the formulation of hypotheses for the possible divergences.

Keywords: Bifocal Modeling. Computational Simulation. Experimentation. Learning.

LISTA DE ILUSTRAÇÕES

Figura 1 – Etapas da Modelagem Bifocal.....	14
Figura 2 – Interface de um experimento simples no <i>NetLogo</i>	16
Figura 3 – Sistemas discretos e sistemas contínuos.....	25
Figura 4 – Rápida alteração de parâmetros em modelos computacionais	28
Figura 5 – Interação de um agente por meio de sensores e atuadores	31
Figura 6 – Exemplo de modelo baseado em agentes com emergência	34
Figura 7 – Fluxograma do processo de Modelagem Bifocal	35
Figura 8 – Exemplo de modelo criado por meio de Modelagem Bifocal.....	36
Figura 9 – Interface do <i>NetLogo</i>	41
Figura 10 – Ambiente de simulação do <i>NetLogo</i> 3D	42
Figura 11 – <i>Turtles</i> com formas, cores e orientações aleatórias	44
Figura 12 – Modelo de interação entre dois gases.....	44
Figura 13 – Importação de uma imagem como ambiente do <i>NetLogo</i>	46
Figura 14 – Estrutura básica do código de um modelo do <i>NetLogo</i>	49
Figura 15 – Condução de calor em gases, líquidos e sólidos	55
Figura 16 – Malha das diferenças finitas	57
Figura 17 – Projeto do experimento de condução.....	60
Figura 18 – Experimento de condução montado.....	61
Figura 19 – Localização dos sensores de temperatura	62
Figura 20 – Teste dos sensores realizado em temperatura ambiente.....	64
Figura 21 – Diagrama dos componentes eletrônicos do experimento.....	64
Figura 22 – Funções integradas ao código do Arduino Uno	65
Figura 23 – Interface do experimento de condução de calor.....	67
Figura 24 – Monitores de temperatura do modelo.....	68
Figura 25 – Diagrama de procedimentos do modelo de condução de calor	69
Figura 26 – Comparação dos dados físicos e computacionais nos pontos A e B	74
Figura 27 – Comparação dos dados físicos e computacionais nos pontos C e D	74
Figura 28 – Comparação dos dados físicos e computacionais no ponto E	75
Figura 29 – Comparação dos dados em A e B com experimento reformulado	77
Figura 30 – Comparação dos dados em C e D com experimento reformulado.....	78
Figura 31 – Comparação dos dados em E com experimento reformulado	78
Figura 32 – Dados experimentais e computacionais após fator de correção	80

LISTA DE QUADROS

Quadro 1 – Exemplos de agentes	33
Quadro 2 – Ferramentas de modelagem computacional	39
Quadro 3 – Código para o cálculo da temperatura.....	66
Quadro 4 – Cálculo de temperatura reformulado	79

LISTA DE TABELAS

Tabela 1 – Medida para calibração dos sensores	71
Tabela 2 – Dados de temperatura coletados no experimento físico.....	72
Tabela 3 – Dados de temperatura coletados na simulação computacional	73
Tabela 4 – Dados de temperatura coletados no experimento físico reformulado.....	76
Tabela 5 – Dados de temperatura coletados na simulação computacional	77
Tabela 6 – Dados da simulação com o fator de correção	80
Tabela 7 – Testes de hipótese da medida 1	81
Tabela 8 – Testes de hipótese da medida 2	82
Tabela 9 – Testes de hipótese da medida 3	82
Tabela 10 – Testes de hipótese da medida 4.....	83
Tabela 11 – Dados do ponto A emparelhados	83
Tabela 12 – Teste t pareado para os dados do ponto A	84
Tabela 13 – Dados do ponto B emparelhados	84
Tabela 14 – Teste t pareado para os dados do ponto B	84
Tabela 15 – Dados do ponto C emparelhados	85
Tabela 16 – Teste t pareado para os dados do ponto C	85
Tabela 17 – Dados do ponto D emparelhados	85
Tabela 18 – Teste t pareado para os dados do ponto D	86
Tabela 19 – Dados do ponto E emparelhados	86
Tabela 20 – Teste t pareado para os dados do ponto E	86
Tabela 21 – Lista de partes do experimento de condução de calor	102
Tabela 22 – Condutividade térmica de materiais à temperatura ambiente	125

LISTA DE ABREVIATURAS E SIGLAS

ABM	Agent Based Modeling
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
LITE	Laboratório de Inovação Tecnológica na Educação
LATEC	Laboratório de Pesquisa Tecnológica em Engenharia
MEC	Ministério da Educação
MBA	Modelagem Baseada em Agentes
MHS	Movimento Harmônico Simples
UNIVALI	Universidade do Vale do Itajaí

LISTA DE SÍMBOLOS

k	Condutividade térmica [W/m.K]
α	Difusividade térmica [m ² /s]
q''	Fluxo de calor [W]
\dot{e}	Geração de energia [W/m ²]
T	Temperatura [K] [°C]

SUMÁRIO

1 INTRODUÇÃO.....	13
1.1 PROBLEMA DE PESQUISA.....	16
1.1.1 Solução Proposta	17
1.1.2 Delimitação de Escopo.....	17
1.1.3 Justificativa	17
1.2 OBJETIVOS	18
1.2.1 Objetivo Geral	18
1.2.2 Objetivos Específicos.....	18
1.3 METODOLOGIA.....	18
1.4 ESTRUTURA DO TRABALHO	19
2 FUNDAMENTAÇÃO TEÓRICA.....	20
2.1 APRENDIZAGEM PRÁTICA.....	20
2.2 MODELAGEM DE SISTEMAS	23
2.3 MODELAGEM BASEADA EM AGENTES.....	30
2.4 MODELAGEM BIFOCAL	34
2.5 FERRAMENTAS DE MODELAGEM E SIMULAÇÃO COMPUTACIONAL	39
2.5.1 NetLogo	40
2.5.2 Validade do <i>NetLogo</i> como ferramenta de modelagem computacional ..	49
2.6 CONDUÇÃO TÉRMICA	54
3 DESENVOLVIMENTO	59
3.1 EXPERIMENTO FÍSICO DE CONDUÇÃO TÉRMICA.....	59
3.2 MODELO COMPUTACIONAL DE CONDUÇÃO TÉRMICA	65
3.3 METODOLOGIA ESTATÍSTICA	70
4 RESULTADOS E DISCUSSÃO.....	71
4.1 TESTES DE HIPÓTESE INDEPENDENTES.....	81
4.2 TESTES DE HIPÓTESE PAREADOS.....	83
5 CONCLUSÃO	87
5.1 SUGESTÕES PARA TRABALHOS FUTUROS.....	88
6 REFERÊNCIAS	89
APÊNDICES	95
ANEXOS	125

1 INTRODUÇÃO

Na área de Engenharia, o raciocínio analítico e a competência para a resolução de problemas devem ser desenvolvidos ao longo da formação do estudante, proporcionando as ferramentas necessárias para que se crie soluções inovadoras e eficientes. (MACHADO; LUZ; PAIVA, 2015, p. 104). Percebe-se, no entanto, falhas no processo de ensino, sendo que uma das causas dessas falhas deriva do fato de que o ensino da técnica de resolução, quando não associado ao ensino do raciocínio lógico ou de métodos que promovam o entendimento do fenômeno físico estudado, provoca dificuldade de compreensão e aprendizado entre os alunos dessa área. (BELHOT; OLIVEIRA NETO, 2006, p. 1).

As disciplinas de física são parte importantíssima do currículo dos cursos de Engenharia, formando a base para a compreensão dos fenômenos físicos estudados posteriormente durante as disciplinas específicas de cada curso. (ROSA; ARAÚJO, 2016, p. 4). Todavia, conforme indicam as autoras ao analisarem publicações na área de ensino em Engenharia, o ensino dessas disciplinas apresenta dificuldades oriundas da falta do desenvolvimento de um pensamento analítico por parte dos alunos necessário à aprendizagem desse conteúdo.

Entretanto, Rosa e Araújo (2016, p. 4) percebem ainda uma tentativa de inclusão de novas metodologias de ensino com o objetivo de tornar o conteúdo das disciplinas mais dinâmico e facilitar o processo de aprendizagem. Por meio dessas informações torna-se evidente que o processo de ensino no campo da Engenharia pode ser melhorado e que o uso de elementos que contribuam para isso deve ser incentivado.

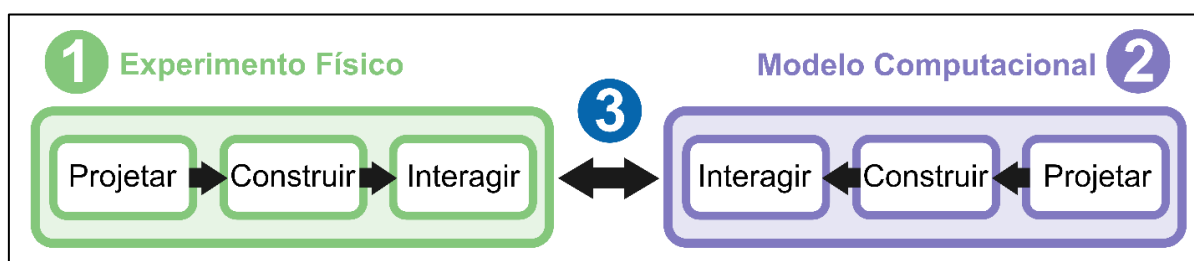
Indo ao encontro dessas ideias, Blikstein *et al.* (2012, p. 296, tradução própria), dizem que “cientistas estão cada vez mais usando modelos como ferramentas importantes para raciocinar, explicar e prever fenômenos no mundo.” Esses modelos, conforme Garcia (2013, p. 19), podem ser físicos, como no caso de protótipos, ou matemáticos, sendo uma representação abstrata da realidade através de equações. O autor ainda diz que a equação ou conjunto de equações que fazem parte do modelo é uma aproximação do processo real, sofrendo diversas simplificações para que o sistema físico seja possível de ser modelado.

Além das etapas de modelagem é importante simular computacionalmente o comportamento do modelo estudado, ou seja, analisar o desenvolvimento comportamental de tal modelo ao longo do tempo, manipulando as variáveis dinamicamente. Isso permite verificar e alterar rapidamente características do fenômeno ou do sistema modelado, algo que seria caro e demandaria tempo no mundo físico. (GARCIA, 2013).

A modelagem e a simulação são usadas nas mais diversas áreas, desde saúde a exploração espacial, o que evidencia suas utilidades como ferramentas de análise e desenvolvimento de conhecimento e aprendizado. Dessa maneira, nos últimos anos foi criada uma técnica chamada Modelagem Bifocal, combinando a experimentação no mundo físico e em ambiente computacional. Um componente central na Modelagem Bifocal é que os estudantes interajam não só na construção de modelos físicos e computacionais, mas conectando e investigando as semelhanças e as diferenças entre esses dois tipos de modelos. (BLIKSTEIN *et al.*, 2012, p. 296).

É possível observar na Figura 1 os passos que fazem parte da Modelagem Bifocal, como determinado por Blikstein *et al.* (2012). Inicialmente deve-se desenvolver um experimento físico relacionado ao fenômeno estudado, como pode ser visto na etapa 1. Após isso é necessário produzir um modelo computacional desse fenômeno, etapa 2, para então comparar os resultados do experimento físico aos resultados da simulação computacional, o que constitui a etapa 3. Em caso de divergência dos resultados, o modelo computacional e o experimento físico devem ser revisados em busca do motivo.

Figura 1 – Etapas da Modelagem Bifocal



Fonte: adaptado de Blikstein *et al.* (2012).

Existem diversos *softwares* de ampla utilização no mercado e que podem ser usados para modelagem e simulação de fenômenos físicos e processos, como por exemplo o *Autodesk Inventor*, o *MATLAB* e o *Wolfram System Modeler*, porém estes

não são gratuitos. Além disso, características como curva de aprendizado e necessidade de capacidade de processamento limitam seu uso como *softwares* de modelagem no ambiente acadêmico. No entanto, os *softwares* citados possuem equipes de desenvolvimento dedicadas a validar os modelos criados com base nos fenômenos físicos que representam, o que permite que sejam usados com elevado grau de confiabilidade.

Sendo assim, ainda dentro desse âmbito, existem os chamados “*Softwares Livres*¹ de código aberto”, ou seja, programas que permitem o uso para qualquer princípio, modificação e distribuição, já que seus desenvolvedores liberam o acesso ao código-fonte do *software* para quem porventura desejar acessá-lo. (XAVIER, 2011). No contexto da Modelagem Bifocal, um dos *softwares* mais relevantes é o *NetLogo*, criado por Wilensky (1999) e inspirado na linguagem “*Logo*”².

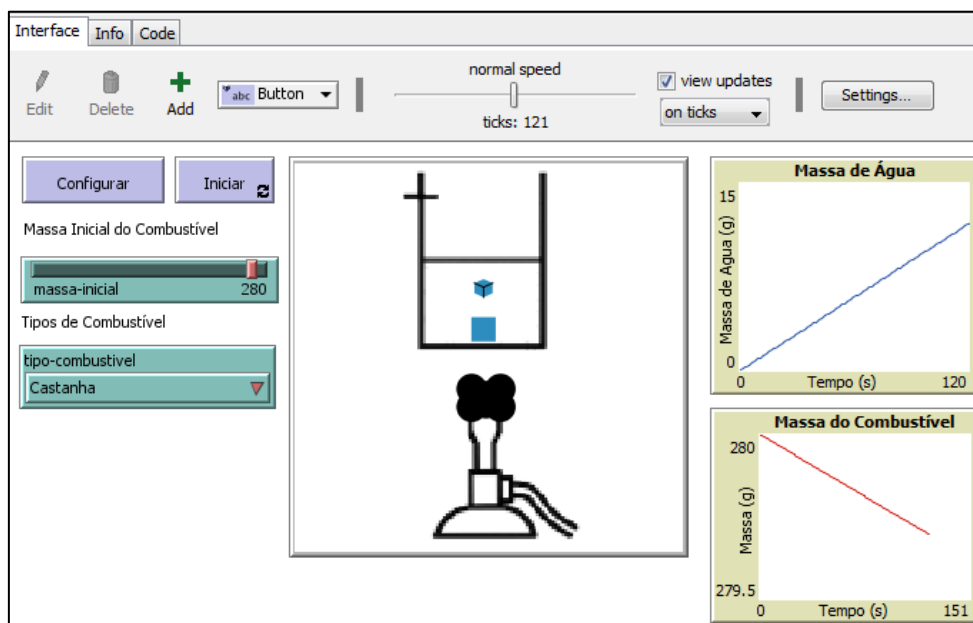
O *NetLogo*, disponível para diversos sistemas operacionais, permite a modelagem através do “paradigma de multiagentes”, no qual cada elemento ou agente do programa, chamado de “*turtle*”, possui regras simples de programação e, no momento da simulação do modelo, um padrão de comportamento mais complexo começa a emergir, representando o fenômeno modelado.

A Figura 2, por sua vez, exemplifica a interface do *NetLogo*. Podem ser adicionados botões, *sliders*, *choosers*, entre outros métodos de entrada e manipulação de variáveis. Da mesma forma, podem ser adicionados elementos de saída, como gráficos e monitores de dados.

¹ Instituições adotam *Softwares Livres* desejando redução de custos e muitas vezes contribuindo com correções de *bugs* e melhorias para o projeto original.

² Concebida em 1967 como uma linguagem de programação educacional.

Figura 2 – Interface de um experimento simples no *NetLogo*



Fonte: LITE (2017).

Diante desse contexto, este trabalho se propõe a contribuir na criação de um modelo computacional de código aberto do fenômeno físico de condução de calor através de Modelagem Bifocal para fins educacionais do curso de Engenharia Mecânica.

As próximas seções deste documento tratam de informar sobre o problema de pesquisa, os objetivos com o desenvolvimento do trabalho e a metodologia utilizada.

1.1 PROBLEMA DE PESQUISA

Streveler *et al.* (2008) consideram que o aprendizado de conceitos fundamentais na Engenharia como forças, troca de calor e corrente elétrica apresenta desafios substanciais aos estudantes. Já de acordo com Silva, Vellasco e Oliveira (2003), os alunos de graduação nos cursos de Engenharia apresentam uma dificuldade marcante em relação à modelagem computacional de problemas. Assim, a busca de metodologias que facilitem o aprendizado deve ser incentivada.

Pepplow, Silva e Sanches (2010, p. 2) acrescentam como uma dificuldade na área de modelagem a escolha do grau de simplificação do modelo em relação ao sistema físico. Essas simplificações podem produzir modelos incorretos e que não apresentam convergência com o fenômeno físico.

A partir disso surgem duas questões relevantes a este trabalho:

1. O *software NetLogo* permite a criação de modelos computacionais para disciplinas de Engenharia Mecânica segundo os requisitos de *software* estipulados por Law e Kelton (2000)?
2. Qual a convergência do modelo implementado em relação ao fenômeno de condução de calor?

1.1.1 Solução Proposta

A busca de ferramentas de modelagem computacional adequadas deve levar em conta critérios de custo, de facilidade e de desempenho. Dessa forma, para que a primeira pergunta seja respondida, deve ser feita uma análise das características do *software NetLogo* através de informações de sua biblioteca, disponível em Wilensky (1999), assim como uma investigação na literatura disponível sobre as aplicações do *NetLogo* à modelagem de fenômenos físicos da Engenharia Mecânica.

Já para responder à segunda pergunta, as seguintes hipóteses foram definidas:

- H0: O modelo implementado apresenta convergência em relação aos experimentos realizado no mundo físico.
- H1: O modelo implementado apresenta divergência em relação ao experimento realizado no mundo físico.

1.1.2 Delimitação de Escopo

Neste trabalho foi criado um modelo computacional de interesse acadêmico para o ensino e aprendizado do fenômeno de condução de calor. É importante deixar claro que este trabalho não contempla a criação de *softwares* de modelagem, preocupando-se apenas com a criação e verificação da validade do modelo estudado, e que várias simplificações foram necessárias quanto à física dos modelos digitais.

1.1.3 Justificativa

Para Blikstein e Wilensky (2007, p. 1, tradução própria), “um caminho poderoso para aplicar tecnologia na melhoria da educação é trazer as ferramentas mais

avançadas disponíveis em laboratórios de pesquisa e adaptá-las para uso em ambientes de ensino.”

Partindo desse pensamento, percebe-se na Modelagem Bifocal e na interação com simulações computacionais de fenômenos físicos uma oportunidade de proporcionar aos estudantes dos cursos de Engenharia, mas não se limitando a eles, uma forma mais interativa de aprendizado. Deseja-se com a introdução de modelos e simulações computacionais transformar o conteúdo teórico e por vezes abstrato de algumas disciplinas em algo prático e tangível, facilmente interpretável.

1.2 OBJETIVOS

Para auxiliar na conclusão deste trabalho, foram definidos os seguintes objetivos.

1.2.1 Objetivo Geral

Criar um modelo computacional do fenômeno físico de condução térmica para uso acadêmico por meio de Modelagem Bifocal.

1.2.2 Objetivos Específicos

1. Verificar a validade do *software NetLogo* como ferramenta de modelagem computacional na Engenharia Mecânica;
2. Definir as variáveis para o modelo;
3. Preparar o experimento no mundo físico;
4. Desenvolver o modelo digital;
5. Conduzir experimentos;
6. Analisar e comparar os principais resultados obtidos nos experimentos.

1.3 METODOLOGIA

Durante a realização deste trabalho foi feita uma revisão bibliográfica, na qual foram utilizados livros, artigos científicos e trabalhos acadêmicos disponíveis no

acervo da universidade ou de aquisição própria. Também foram utilizados trabalhos acadêmicos e informações disponíveis em meio eletrônico para discutir sobre a validade do *software NetLogo* como ferramenta de modelagem no âmbito do curso de Engenharia Mecânica.

Já na fase de coleta de dados foram utilizados equipamentos disponíveis no Laboratório de Física da Universidade do Vale do Itajaí, como termômetros e fontes elétricas, e equipamentos de construção própria para que os resultados dos experimentos no mundo físico pudessem ser comparados aos resultados do modelo proposto em ambiente computacional.

A presente pesquisa pode ser classificada como de finalidade aplicada, gerando ao fim de sua execução produtos sob a forma de um modelo computacional associado a um experimento físico. Quanto aos procedimentos desta pesquisa, pode-se dizer que é do tipo experimental, de abordagem mista, contendo elementos qualitativos e elementos quantitativos. A pesquisa experimental “[...] consiste em determinar um objeto de estudo, selecionar as variáveis que seriam capazes de influenciá-lo, definir as formas de controle e de observação dos efeitos que a variável produz no objeto.” (GIL, 2002, p. 47).

1.4 ESTRUTURA DO TRABALHO

A seção 1 deste trabalho apresenta os principais conceitos relacionados ao problema de pesquisa, assim como detalha os objetivos do trabalho e a metodologia de execução das atividades para que esses objetivos fossem cumpridos.

Já a seção 2 entra em mais detalhes sobre a Modelagem de Sistemas e sobre o método de Modelagem Bifocal, passando também pela verificação da validade do *NetLogo* como *software* de modelagem e detalhando o fenômeno físico do qual foi criado o modelo computacional.

A seção 3 aborda o desenvolvimento do trabalho, apresentando os experimentos realizados. Os resultados obtidos são apresentados na seção 4, assim como uma discussão sobre esses dados.

Por fim, na seção 5 são feitas as considerações finais do autor, comentando os principais resultados obtidos e deixando sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção são apresentados os principais conceitos teóricos relacionados ao tema do trabalho. Inicialmente o tópico de aprendizagem prática é discutido com enfoque na área de Engenharia. Após isso, os conceitos de Modelagem de Sistemas são apresentados, com ênfase em modelos computacionais. A seguir o método de Modelagem Bifocal é detalhado, sendo seguido pela verificação da validade do *NetLogo* como ferramenta de modelagem computacional no contexto da Engenharia Mecânica e pelos componentes teóricos do fenômeno físico de condução de calor.

2.1 APRENDIZAGEM PRÁTICA

As áreas do conhecimento relacionadas ao curso de Engenharia Mecânica, segundo o Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq (2013), compreendem o estudo de Fenômenos de Transporte, Engenharia Térmica, Mecânica dos Sólidos, Projeto de Máquinas e Processos de Fabricação. Cada um desses itens contém diversas subáreas, 24 quando somadas.

Percebe-se, a partir dessa informação, que esse curso apresenta uma grande área de estudo, possibilitando que o profissional formado escolha entre diversos campos de atuação, porém exigindo que o estudante se familiarize com um enorme leque de conteúdos apresentados durante as atividades formativas. Isso também é evidenciado por UNIVALI (2018), que lista as disciplinas obrigatórias oferecidas pelo curso de Engenharia Mecânica da Universidade do Vale do Itajaí.

Ainda assim, os cursos de Engenharia promovem um método de ensino considerado por Casale (2013, p. 19) como tradicional, em muitos casos não proporcionando atividades práticas de maneira satisfatória em relação à teoria e às necessidades do profissional, e isolando o conhecimento de cada subárea, de forma a não promover plenamente uma interação entre as diversas disciplinas do curso.

De acordo com esse pensamento, a forma com que os conteúdos são transmitidos ao acadêmico de Engenharia não proporciona o desenvolvimento das competências críticas do aluno de maneira satisfatória, como também o raciocínio necessário para a posterior atuação do profissional no mercado de trabalho. (CASALE, 2013).

A necessidade de desenvolvimento das competências críticas e do raciocínio autônomo dos alunos é tão importante que é evidenciada pelas diretrizes gerais do Ministério da Educação – MEC, diretrizes essas que são listadas por Casale (2013, p, 28) a fim de evidenciar a importância da promoção de atividades interativas no meio universitário. São elas:

- Domínio de linguagens;
- Compreensão de fenômenos;
- Construção de argumentações;
- Solução de problemas;
- Elaboração de propostas.

A autora ainda alega que métodos indutivos de ensino, baseados na experimentação, estão mais alinhados à formação das competências requisitadas ao profissional de Engenharia, promovendo “a adoção da aprendizagem profunda, desenvolvimento intelectual, pensamento crítico e habilidades de aprendizagem autônoma”. (CASALE, 2013, p. 33).

Em contrapartida a essa deficiência no ensino prático, pode ser percebida uma tentativa por parte de alguns professores e instituições de incluir novas metodologias de ensino acadêmico, inclusive com a possibilidade de utilizar a Aprendizagem Baseada em Problemas – método pedagógico baseado na solução de problemas reais ou fictícios que são apresentados pelo professor antes da teoria a fim de estimular a obtenção de novos conhecimentos – como uma abordagem benéfica para o processo de aprendizagem dos alunos de Engenharia. (CASALE, 2013; ROSA; ARAÚJO, 2016).

Ainda dentro do âmbito da Aprendizagem Baseada em Problemas é possível afirmar que seu objetivo primordial é promover a busca, por parte dos estudantes, da solução dos problemas propostos, “interagindo com seus pares, buscando informações, formulando hipóteses, tomando decisões e emitindo julgamento.” (CASALE, 2013, p. 34). Portanto, o papel do professor nesse caso é de facilitador, orientando da melhor maneira possível o processo de aprendizagem tanto do indivíduo quanto do grupo como um todo.

Com a mesma intenção e de maneira semelhante, Queiroz, Lima e Cerqueira (2017, p. 2) citam a importância de uma readequação do processo de ensino para uma melhor formação de competências através da metodologia conhecida como “espaço-tempo”. Nesta, o estudante é tratado como um profissional atuante, que deve criar soluções para problemas e nesse processo toma conhecimento do funcionamento das soluções existentes.

Logo, o papel das atividades práticas neste contexto é crucial para proporcionar aos estudantes ferramentas e meios de explorar, discutir e testar hipóteses sobre o assunto de estudo, adquirindo, durante o processo, conhecimento técnico e competências para a atuação posterior na vida profissional.

Além dos métodos supracitados, outros tantos métodos e ferramentas de auxílio para o processo de aprendizagem têm se destacado nos últimos anos. Um grande exemplo disso é o processo de prototipagem que foi vastamente facilitado pela técnica de impressão 3D, como também a área da robótica que foi beneficiada pela criação de *kits* eletrônicos como o *Arduino* e o *Raspberry Pi*. Avanços esses que, especificamente, têm a capacidade de propiciar a democratização do conhecimento obtido pelos graduandos em Engenharia. (BERMAN, 2012; YAMANOOR; YAMANOOR, 2017)

De forma concreta é possível evidenciar as atividades práticas que a Universidade do Vale do Itajaí (UNIVALI) proporciona aos alunos de Engenharia Mecânica, como no caso dos laboratórios específicos do curso onde são realizados ensaios metalográficos, interação com componentes de robótica, impressão de componentes através de manufatura aditiva e estudos envolvendo fluidos, além do Laboratório de Pesquisa Tecnológica em Engenharia (LATEC) que, por sua vez, disponibiliza seu espaço e equipamentos para estudos de soldagem, de usinagem e também de hidráulica. (PEDRO, 2013, p. 40).

É imprescindível ter em mente que para a obtenção de uma experiência plena de aprendizado envolvendo atividades práticas se faz necessário dispor de laboratórios e equipamentos adequados e pertinentes ao curso, em bom estado de uso e conservação. Entretanto, no cenário real, muitas vezes tais laboratórios e equipamentos não apresentam todos esses pré-requisitos para o melhor desenvolvimento do aprendizado prático. (GUILLERMO, 2016, p. 21).

Nas últimas décadas, pela democratização do ensino superior e a consequente entrada de estudantes de classes sociais menos favorecidas nos cursos de Engenharia, o perfil do estudante dessa área mudou. Muitos desses estudantes necessitam trabalhar durante a graduação, o que implica em uma conciliação das atribuições profissionais e acadêmicas. Logo, é importante ressaltar que em muitos casos o próprio estudante não dispõe tempo para realizar atividades práticas fora dos horários de aula.

Essas informações evidenciam a importância de proporcionar meios alternativos e democráticos de experimentação aos estudantes, como a interação em ambiente virtual com modelos computacionais válidos, de maneira que tenham a possibilidade de aprimorar seu conhecimento sobre o assunto de estudo.

Portanto, a lacuna do aprendizado prático na Engenharia é amplamente conhecida na mesma proporção que a importância desse tipo de aprendizado é demonstrada. É preciso, cada vez mais, que professores e universidades promovam o uso de novos métodos e ferramentas que viabilizem as atividades práticas para que os estudantes de Engenharia obtenham o melhor desenvolvimento possível de suas competências profissionais e cheguem ao mercado de trabalho com o *know-how* necessário para qualificá-los como profissionais completamente aptos.

2.2 MODELAGEM DE SISTEMAS

Modelagem de Sistemas é o campo de estudo que busca dar subsídio para a construção e análise de representações da realidade. Essas representações são construídas com objetivos definidos, como compreender melhor o comportamento de um processo, obter informações sem alterar o sistema real ou alterar variáveis para testar hipóteses sobre o sistema. (SOARES, 1992, p. 2). Dentro da área de Modelagem de Sistemas estão presentes alguns conceitos de extrema relevância, como sistemas, modelos e simulações, que são apresentados a seguir.

De forma concisa, um sistema pode ser definido como o conjunto de itens que interagem para atingir um determinado objetivo e em que exista uma relação de causa e efeito. Já no contexto da Modelagem de Sistemas, sistema é o processo ou o fenômeno de interesse, o objeto de estudo, do qual se deseja extrair informações ou algum conhecimento. (SOARES, 1992, p. 2; LAW; KELTON, 2000, p. 1).

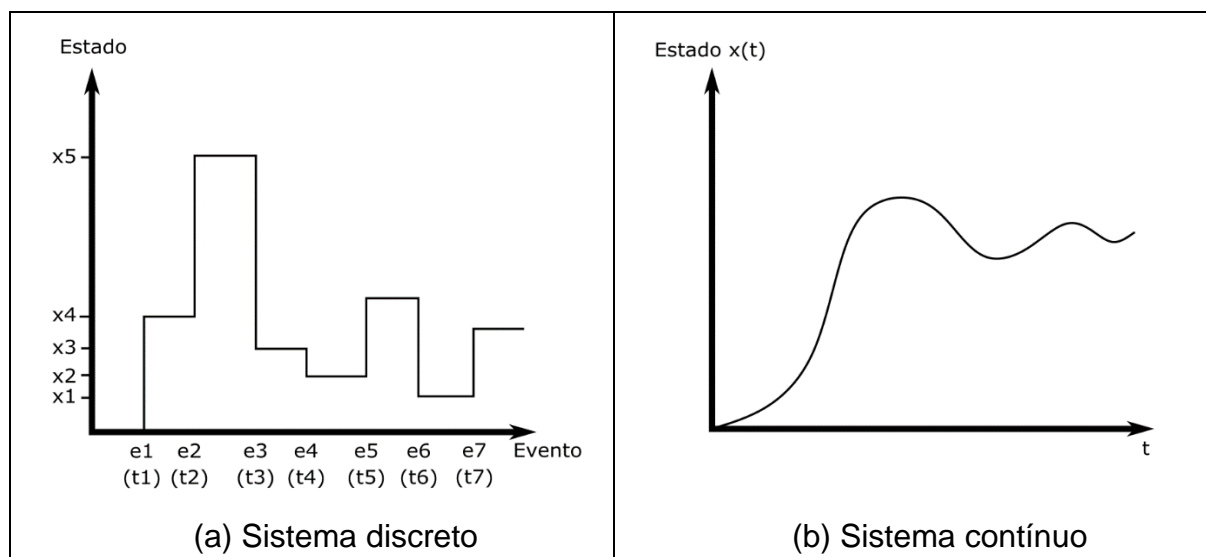
No entanto, Law e Kelton (2000, p. 3) afirmam que o que se entende por sistema pode variar dependendo do objetivo ou foco do estudo, já que o conjunto de elementos de interesse em um estudo pode ser apenas um subconjunto dos elementos de interesse para um outro estudo. Nesse sentido, entender a extensão do sistema e as relações com o meio no qual ele está contido é importantíssimo para que se possa analisá-lo corretamente.

Os sistemas se caracterizam pela forma com que se comportam no decorrer tempo, podendo ser estáticos ou dinâmicos e, neste último caso, ainda podem ser discretos ou contínuos. Pode-se dizer que sistemas estáticos não apresentam alteração de variáveis ao longo do tempo, como em uma viga que sofre esforços mecânicos constantes. Já os sistemas dinâmicos apresentam variação ao longo do tempo, com mudanças em parâmetros e em variáveis. Um exemplo clássico desse tipo de sistema é o nível de fluido que varia em um tanque de acordo com a taxa de entrada e a taxa de saída de fluido. (CARVALHO, 2003, p. 22).

Ainda no âmbito dos sistemas dinâmicos, existem os sistemas discretos, compreendendo processos em que os eventos ocorrem em instantes específicos de tempo, como é o caso de processos nos quais há a criação de filas e que podem ser representados de maneira discreta. Já os sistemas contínuos são aqueles em que eventos ocorrem em todos os instantes de tempo, ou ainda, as mudanças de estado ocorrem a cada fração de tempo, continuamente. (LAW; KELTON, 2000, p. 3; CARVALHO, 2003, p. 22-23).

A Figura 3 evidencia as diferenças entre esses dois casos, mostrando as características citadas tanto nos sistemas discretos (a) quanto nos sistemas contínuos (b).

Figura 3 – Sistemas discretos e sistemas contínuos



Fonte: Adaptado de Carvalho (2003, p. 23)

A partir dos conceitos de sistema apresentados, pode-se avançar um passo e partir para o entendimento de modelos. Soares (1992, p. 3) compreende modelos como descrições de um sistema. Já Eykhoff (1974, *apud* GARCIA, 2013, p. 19) define que um modelo “é uma representação dos aspectos essenciais de um sistema, que apresenta conhecimento desse sistema em uma forma utilizável.”

De qualquer modo, percebe-se nessas definições que o modelo não é uma cópia exata e idêntica do sistema que busca representar, mas sim uma abstração, uma simplificação realizada de maneira que é necessário apenas representar as características relevantes ao estudo e as variáveis que possivelmente podem alterar essas características. Sobre isso, Garcia (2013, p. 19) adverte que a relação entre o custo para se obter um modelo e o nível de detalhes desejado deve ser verificada para que a criação do modelo não se torne inviável.

Blikstein e Wilensky (2007) dizem que “um passo necessário para a construção de modelos é encontrar as regras elementares dentro de um sistema.” Então, é preciso que seja feita uma análise minuciosa, caso a caso, do sistema de interesse, a fim de corretamente extrair e representar essas características fundamentais, e selecionando, isolando e identificando corretamente as variáveis de entrada e saída do sistema.

Os autores que tratam da teoria de Modelagem de Sistemas dividem os modelos em alguns grupos. Primeiramente, é possível classificá-los usando como

critério a continuidade, uma vez que os modelos, de maneira análoga aos sistemas, podem ser criados como discretos ou contínuos, o que depende do sistema original e da aplicação desejada. (LAW; KELTON, 2000, p. 6).

Não é necessário que um sistema contínuo seja representado continuamente, assim como não é necessário que um sistema discreto seja representado discretamente. É o que afirmam Law e Kelton (2000, p. 6) quando explicam que “a decisão de se usar um modelo discreto ou contínuo para um sistema em particular depende dos objetivos específicos do estudo.”

Também é possível classificar modelos quanto à forma de construção, de modo que modelos podem ser físicos, matemáticos ou computacionais, sendo que estes últimos apenas tornaram-se realizáveis devido ao aumento do poder computacional visto nas últimas décadas. (KELTON; SADOWSKI; ZUPICK, 2015).

Modelos físicos, também conhecidos como modelos icônicos, são aqueles construídos fisicamente com materiais reais, normalmente em escala reduzida. (LAW; KELTON, 2000, p. 4). Para que o estudo desses modelos seja útil, realizam-se simulações de modelos físicos envolvendo a introdução do modelo a condições que representem seu funcionamento real. Como exemplo disso é possível citar as miniaturas em escala de navios construídas com o propósito de estudar o fluxo de água pelo casco e o desempenho da embarcação a ser construída.

Já os modelos matemáticos, ou modelos simbólicos, são representações de sistemas por meio de uma série de equações matemáticas, “através das quais o comportamento do sistema pode ser obtido pela atribuição de valores aos parâmetros do modelo e a solução de equações” (SOARES, 1992, p. 3). Algo a ser notado é que caso o modelo matemático seja simples, é possível manipular as equações de modo que se obtenha uma solução analítica, um valor exato para o problema estudado. (LAW; KELTON, 2000, p. 5).

Por fim, existe a possibilidade de simular computacionalmente o comportamento de sistemas desde os mais simples aos mais complexos através da criação de modelos computacionais, permitindo que as variáveis de entrada do modelo sejam manipuladas em tempo real e assim produzindo resultados próximos aos do sistema físico. (LAW; KELTON, 2000, p. 5).

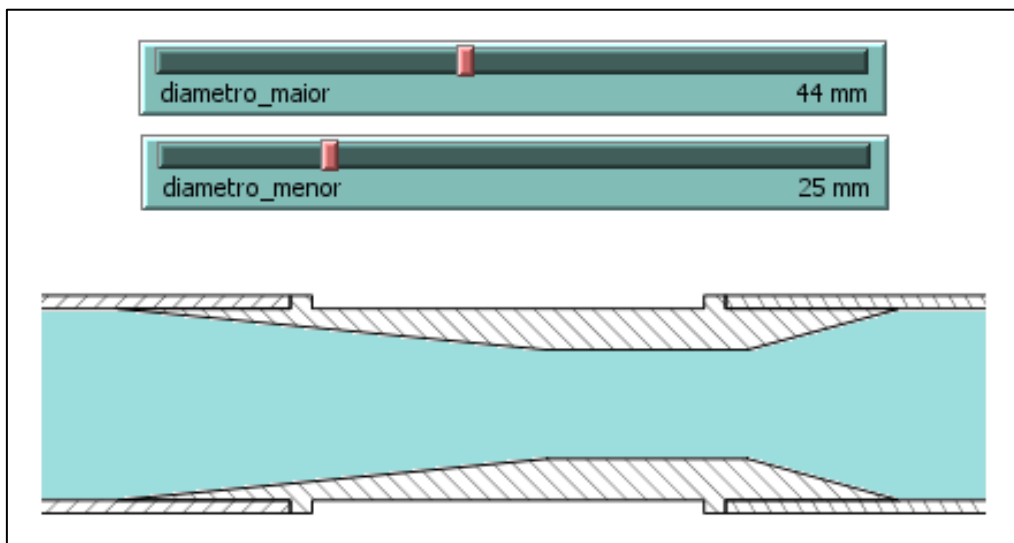
Nesse sentido, Wilensky e Rand (2015, p. 22, tradução própria) definem modelo computacional como “um modelo que recebe certos valores de entrada, manipula estas entradas de maneira algorítmica, e gera saídas.” Já simulações computacionais permitem que os mais diversos modelos sejam estudados “[...] por avaliação numérica usando *softwares* desenhados para imitar as características e operações do sistema, muitas vezes através do tempo” (KELTON; SADOWSKI; ZUPICK, 2015, p. 5).

Os modelos computacionais podem ser criados em *softwares* especializados, como o *NetLogo*, *Arena Simulator*, *Autodesk Inventor*, ou até mesmo em linguagens de propósito geral, como *C++*, *Java* e *Swift*. Cada um apresentando vantagens e desvantagens em determinadas aplicações, sendo que a escolha do *software* no qual o modelo será criado depende fortemente do tipo e características do sistema, assim como do objetivo da análise. (KELTON; SADOWSKI; ZUPICK, 2015, p. 10).

Desse modo, as simulações de modelos computacionais facilitam a análise de forma que os parâmetros de execução podem ser alterados dinamicamente, ou de maneira rápida, sem custo econômico, e facilitando a análise da influência de cada parâmetro no processo ou sistema, diferentemente dos modelos físicos, que necessitam muitas vezes de uma reformulação estrutural para que parâmetros sejam alterados.

Como exemplo dessa discrepância entre as simulações e os modelos físicos é possível citar o caso de uma modelagem do fluxo de água pelo interior de uma tubulação. Em uma simulação computacional pode-se rapidamente alterar os parâmetros de diâmetro, como é mostrado na Figura 4, de modo que os efeitos dessa mudança ocorram no mesmo momento. Já em um modelo físico seria necessário substituir completamente a tubulação, demandando um tempo maior para sua realização e, conseqüentemente, acarretando em maiores custos.

Figura 4 – Rápida alteração de parâmetros em modelos computacionais



Nesse exemplo seria possível que o estudante compreendesse mais sobre o comportamento dos fenômenos de fluxo de fluidos e transporte de massa através da interação com o modelo computacional, de modo que ao variar o parâmetro de diâmetro, características como pressão e velocidade do fluido seriam modificadas como consequência, facilitando a rápida visualização dessas mudanças de estado através de gráficos ou representações visuais.

Outra vantagem dos modelos computacionais, de acordo com Kelton, Sadowski e Zupick (2015, p. 6), é a possibilidade de poder lidar com modelos de sistemas com muitas variáveis simultâneas, sem que sejam necessárias muitas simplificações que poderiam prejudicar a validade do modelo.

Porém, esse tipo de modelo precisa ser extensivamente testado e comparado ao fenômeno real para que tenha validade e utilidade. Logo, neste trabalho o modelo computacional será construído de forma conjunta ao experimento físico análogo para que seja feita a comparação e a validação, como determina a Modelagem Bifocal, melhor detalhada na seção 2.4.

Kelton, Sadowski e Zupick (2015, p. 560) citam que a validade de um modelo está ligada à “tarefa de garantir que o modelo se comporte como o sistema real.” Law e Kelton (2000, p. 265) dizem que para um sistema complexo, definido por Wilensky e Rand (2015, p. 23) como um “sistema composto de muitas partes interativas distribuídas”, esse comportamento nunca será idêntico ao mundo físico, já que a simulação de um modelo pode ser apenas uma aproximação do processo real. Apesar

desse aspecto, um modelo criado de forma que represente corretamente o comportamento do sistema é considerado válido e pode ser usado para tomada de decisões ou para aquisição de informações sobre o sistema real.

No entanto também é preciso levar em consideração os aspectos negativos e/ou as dificuldades quanto à modelagem computacional de sistemas, já que muitos sistemas do mundo real envolvem incertezas e aleatoriedades nas relações de entradas e saídas. Uma forma de lidar com essa questão é criar um modelo probabilístico e executá-lo tantas vezes quanto necessário até que uma resposta aceitável seja obtida estatisticamente. (KELTON; SADOWSKI; ZUPICK, 2015, p. 7).

É sabido que os últimos desenvolvimentos na área de Modelagem de Sistemas estão diretamente ligados ao avanço do poder computacional, à otimização de supercomputadores e ao uso de inteligência artificial, revolucionando o modo como entendemos os sistemas complexos da natureza e ampliando a gama de sistemas possíveis de serem corretamente modelados e operados, como por exemplo na programação de carros autônomos, que leva em conta a modelagem e interação em tempo real do veículo com seu ambiente. (PALMER, 2015; FUJIMOTO *et al.*, 2016, p. 47; FERNANDES *et al.*, 2017).

Atualmente, utilizando muitos dos conceitos apresentados neste trabalho, são feitas desde análises estáticas e dinâmicas da estrutura de pontes e edifícios, conforme pode ser visto em Zhou e Chen (2015) e em Bastos e Silva (2016), até mesmo modelagem e simulação das estruturas que compõe o universo, como no estudo de Pillepich *et al.* (2017). Isso mostra o poder e a versatilidade dessa área de estudo, possibilitando que sistemas complexos e de difícil compreensão sejam estudados.

A modelagem e simulação de sistemas traz também benefícios para a área de Engenharia Mecânica, como por exemplo na forma de modelagem do funcionamento de mecanismos em um sistema mecânico ou mesmo no balanceamento de sistemas de produção, entre muitos outros casos (CAVALCANTI *et al.*, 2005; HOSSEINPOUR; HAJIHOSSEINI, 2009). Essa área possibilitou muitos avanços vistos nas últimas décadas em setores como o da aeronáutica, através da modelagem e simulação de fluidos com alta precisão, transformando as tecnologias e proporcionando o aumento da eficiência dos sistemas no mundo moderno (NASA, 2018).

Com isso em mente, as próximas seções deste trabalho tratam de ferramentas e métodos que podem ser utilizados para a geração de modelos computacionais, principalmente as relacionadas ao contexto da Modelagem Bifocal, tema de estudo da seção 2.4, aplicada à Engenharia Mecânica.

2.3 MODELAGEM BASEADA EM AGENTES

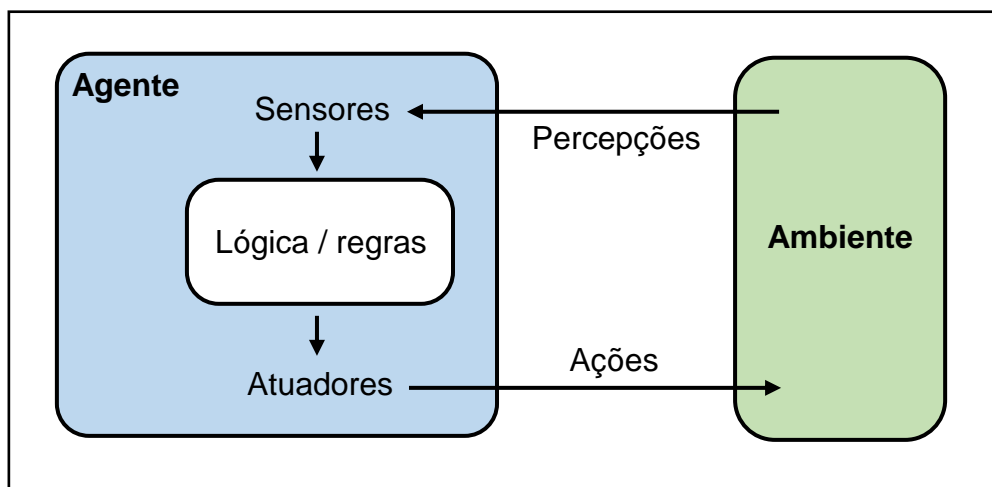
Uma das características marcantes dos trabalhos realizados na área de Modelagem Bifocal é a criação de modelos digitais por meio do paradigma de Modelagem Baseada em Agentes (MBA), do inglês *Agent Based Modeling* (ABM). Esse método de modelagem computacional deriva dos estudos de Inteligência Artificial e permite a manipulação do comportamento de elementos inteligentes, independentes e autônomos, conhecidos como agentes, para representar características de um sistema. (WILENSKY; RAND, 2015; RUSSELL; NORVIG, 2013).

Em síntese, um agente pode ser definido como um objeto ou elemento de simulação computacional que possui autonomia, comportamentos programados através de regras lógicas e matemáticas, estados de operação e propriedades individuais, como cor e tamanho. (WILENSKY; RAND, 2015, p. 22). No mesmo sentido, Russell e Norvig (2013, p. 31) fornecem a definição de que “um agente é tudo o que pode ser considerado capaz de perceber seu ambiente por meio de sensores e de agir sobre esse ambiente por intermédio de atuadores.”

A partir disso, percebe-se que um agente ao ser inserido em um ambiente, seja físico ou computacional, tem capacidade de agir sobre esse ambiente, sobre o próprio agente ou sobre outros agentes, com base em certas percepções sobre seu meio, que são fornecidas por sensores virtuais ou físicos. Sobre essa interação entre agentes e ambientes, Russell e Norvig (2013, p. 31) dizem que “a escolha de ação de um agente em qualquer instante dado pode depender da sequência inteira de percepções recebidas até o momento, mas não de percepções não recebidas”.

A Figura 5 exemplifica os elementos de um agente simplificado conforme citado, mostrando sua relação com o ambiente no qual está imerso, assim como um elemento que representa a lógica ou o mapeamento das ações a serem tomadas de acordo com as percepções do agente.

Figura 5 – Interação de um agente por meio de sensores e atuadores



Fonte: Adaptado de Russell e Norvig (2013, p. 32).

Existem dois conceitos importantes que permeiam o estudo de agentes. O primeiro deles é a medida de desempenho do agente (MD), que é uma indicação do grau de sucesso do agente em determinada tarefa, sendo avaliada a partir do exterior, ou seja, a partir do ambiente. (RUSSELL; NORVIG, 2013, p. 33).

Já o segundo conceito é conhecido como racionalidade, que está ligado à capacidade do agente de, segundo as regras do ambiente, obter o melhor desempenho em determinada tarefa. Nas palavras de Russell e Norvig (2013, p. 33), “Para cada sequência de percepções possível, um agente racional deve selecionar uma ação que se espera venha a maximizar sua medida de desempenho”.

Segundo esses parâmetros, o tipo mais básico de agente é o que segue regras mapeadas em tabelas de ações preestabelecidas, de maneira que uma percepção oriunda dos sensores disponíveis pode ser pesquisada em uma tabela que corresponde cada tipo de percepção a uma determinada ação, fornecendo assim a ação correspondente a ser executada pelo atuador.

Este tipo de agente está se tornando impraticável recentemente devido à enorme quantidade de percepções possíveis a serem mapeadas antecipadamente em sistemas modernos. (RUSSELL; NORVIG, 2013, p. 41). No entanto, para sistemas e modelos com uma grande quantidade de agentes e regras extremamente simplificadas de interação entre agentes e ambientes, esses agentes com parâmetros tabelados ainda podem ser usados por oferecer certa simplicidade de construção. (WILENSKY; RAND, 2015).

Outros tipos de agentes são melhores adaptados para tarefas mais complexas, como agentes reativos simples, agentes reativos baseados em modelo, agentes baseados em objetivos e agentes baseados na utilidade. Cada um desses com características e mecanismos próprios para gerar ações. (RUSSELL; NORVIG, 2013, p. 42).

No contexto do presente trabalho os tipos de agentes mais importantes para a modelagem dos fenômenos físicos escolhidos são os agentes reativos simples e os agentes reativos baseados em modelo, que serão descritos a seguir, porém não se deve ignorar a utilidade dos agentes baseados em objetivos e agentes baseados na utilidade, que compõe o que há de mais elaborado no estudo da Modelagem Baseada em Agentes, de acordo com Russell e Norvig (2013).

Os agentes reativos simples são aqueles que agem com base apenas na percepção atual, não guardando um histórico de seu estado anterior ou ainda usando esse histórico para tomada de decisão. Pode-se dizer que então que esses agentes não possuem uma memória sobre os fatos passados e tomam decisões usando estruturas condicionais do tipo “se X então Y”, predefinidas. (RUSSELL; NORVIG, 2013, p. 42).

Já os agentes reativos baseados em modelo são os que possuem a capacidade de armazenar algum estado anterior, ou então percepções passadas, que permitem ao agente considerar a parte do ambiente que ele não pode observar no momento, porém com certa incerteza, já que os parâmetros de ambiente podem mudar com o passar do tempo. (RUSSELL; NORVIG, 2013, p. 44).

Alguns exemplos de agentes podem ser facilmente identificados no mundo real, como um agente humano, que possui percepções na forma de sentidos, como visão e audição, e pode atuar sobre seu ambiente por meio de membros e extremidades. Outro exemplo, mais relevante para a área computacional e de Engenharia, são robôs automatizados com capacidade de percepção, localização e muitas vezes de movimentação espacial, que usam essas capacidades para movimentar itens a partir de uma programação prévia, efetivamente agindo sobre seu ambiente. (RUSSELL; NORVIG, 2013).

O Quadro 1, disponibilizado por Russell e Norvig (2013), exemplifica alguns agentes que poderiam ser criados do mundo real, identificando as medidas de

desempenho que poderiam ser usadas para que o agente alcance seu objetivo, assim como mostra o ambiente no qual estes agentes estariam situados, os atuadores e os sensores disponíveis a esses agentes.

Quadro 1 – Exemplos de agentes

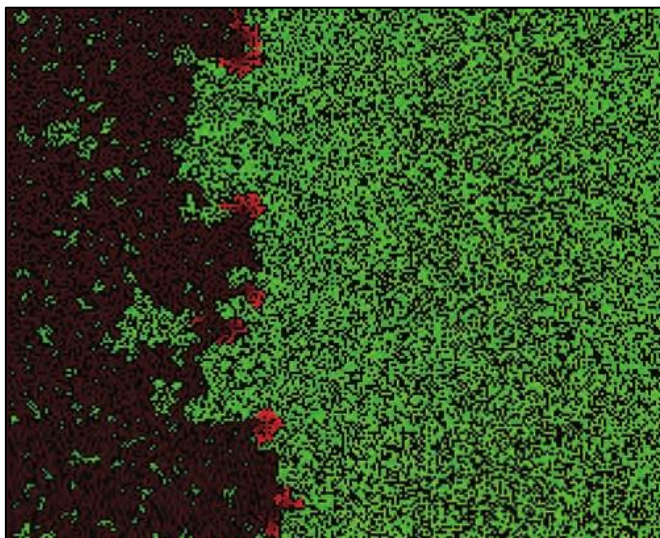
Tipo de agente	Medida de desempenho	Ambiente	Atuadores	Sensores
Sistema de diagnóstico médico	Paciente saudável, minimizar custos	Paciente, hospital, equipe	Exibir perguntas, testes, diagnósticos, tratamentos, indicações	Entrada pelo teclado para sintomas, descobertas, respostas do paciente
Sistema de análise de imagens de satélite	Definição correta da categoria da imagem	Link de transmissão de satélite em órbita	Exibir a categorização da cena	Arrays de pixels em cores
Robô de seleção de peças	Porcentagem de peças em bandejas corretas	Correia transportadora com peças, bandejas	Braço e mão articulados	Câmera, sensores angulares articulados
Controlador de refinaria	Maximizar pureza, rendimento, segurança	Refinaria, operadores	Válvulas, bombas, aquecedores, mostradores	Sensores de temperatura, pressão, produtos químicos
Instrutor de inglês interativo	Maximizar nota de aluno em teste	Conjunto de alunos, ambiente de testes	Exibir exercícios, sugestões, correções	Entrada pelo teclado

Fonte: Adaptado de Russell e Norvig (2013, p. 37)

Uma característica marcante dos modelos que utilizam agentes é a emergência de padrões produzidos através da interação entre a população de agentes, de forma que as regras de programação simples, introduzidas em cada agente, produzem um comportamento complexo quando se observa o modelo como um todo. (WILENSKY; RAND, 2015, p. 6).

Wilensky e Rand (2015, p. 18) fornecem como exemplo de emergência um modelo da propagação de um incêndio florestal por meio da Figura 6. Formalmente, este fenômeno é descrito por meio de equações diferenciais de difícil compreensão. Já nesse modelo as árvores são agentes e o conjunto desses agentes forma uma floresta. A propagação do incêndio de uma árvore para a próxima segue regras simples, mas que produzem um comportamento semelhante ao observado na natureza quando se observa o conjunto como um todo.

Figura 6 – Exemplo de modelo baseado em agentes com emergência



Fonte: Wilensky e Rand (2015, p. 19).

O uso de agentes pode ser incorporado a linguagens de programação de alto nível, como no caso da linguagem *Prolog*, muito usada no contexto da Inteligência Artificial pela simplicidade da sintaxe de sua linguagem (RUSSELL; NORVIG, 2013, p. 296). No entanto, no escopo da Modelagem Bifocal o *software NetLogo* (WILENSKY, 1999) demonstra ser uma linguagem simples para criação de modelos por pesquisadores e estudantes de diversas áreas (TISUE; WILENSKY, 2004).

Como informado no início dessa seção, a Modelagem Baseada em Agentes certamente apresenta utilidade no contexto da Modelagem Bifocal, formando também a base dos modelos criados posteriormente neste trabalho. Logo, em posse do conhecimento sobre agentes e ambientes é possível seguir para o estudo dos conceitos de Modelagem Bifocal e das características do *software NetLogo*.

2.4 MODELAGEM BIFOCAL

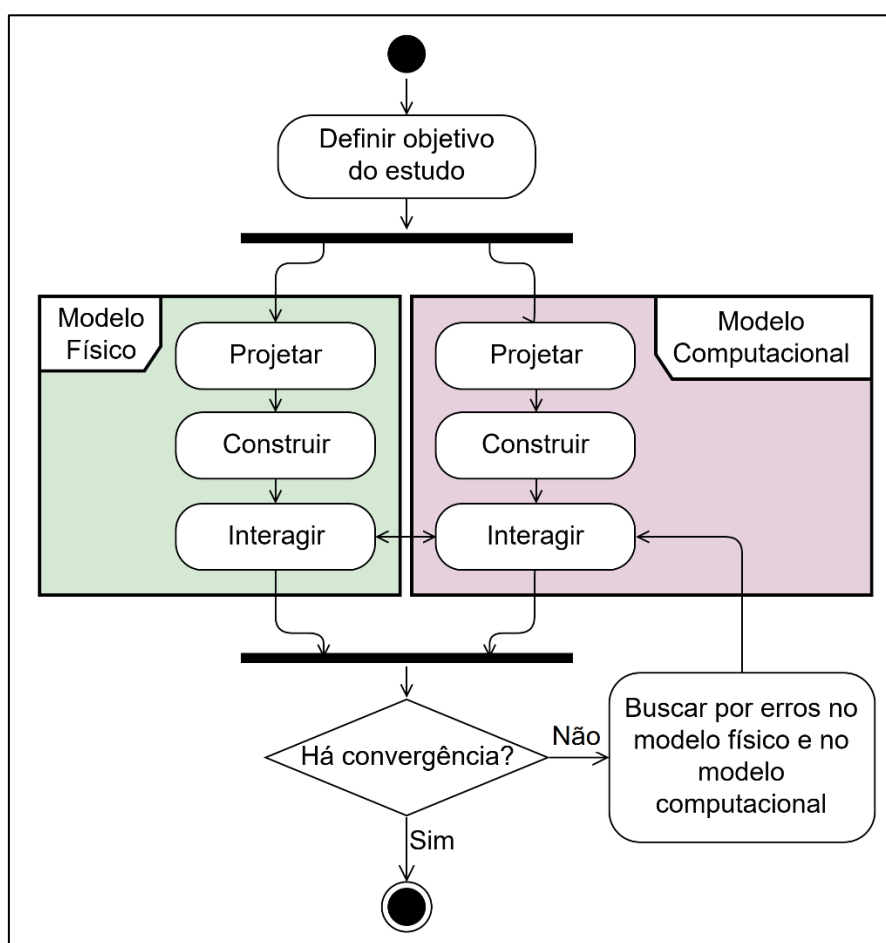
Modelagem Bifocal é um método de modelagem de sistemas que busca unir a construção de modelos digitais à experimentação no mundo físico. Esse método foi desenvolvido por Blikstein e Wilensky (2007, p. 4) como uma forma de promover uma ligação entre teoria e prática e para facilitar o entendimento de determinados fenômenos físicos através da observação e da experimentação.

Blikstein *et al.* (2012, p. 296) enunciam três passos para que se construa um modelo por meio de Modelagem Bifocal. Inicialmente se deve “desenvolver um modelo

físico para explorar um fenômeno científico, usando sensores eletrônicos e uma placa de coleta de dados”. Após isso é preciso construir um modelo computacional do mesmo fenômeno usando um *software* de modelagem computacional. O mesmo autor diz que, por fim, os dois experimentos devem ser executados simultaneamente, de modo que seja possível visualizar e comparar os dados experimentais lado a lado.

O processo descrito é iterativo, já que em caso de divergência nos resultados faz-se necessário buscar o motivo e corrigir o erro, seja no modelo físico ou no modelo computacional, mesmo quando esse erro é uma simplificação feita de forma incorreta, por exemplo, para que então os experimentos sejam executados novamente até que o resultado gerado no experimento computacional se comporte de forma semelhante ao que se observa no experimento físico. Isso pode ser visto na Figura 7, juntamente com os passos mencionados anteriormente.

Figura 7 – Fluxograma do processo de Modelagem Bifocal



Fonte: adaptado de Blikstein *et al.* (2012).

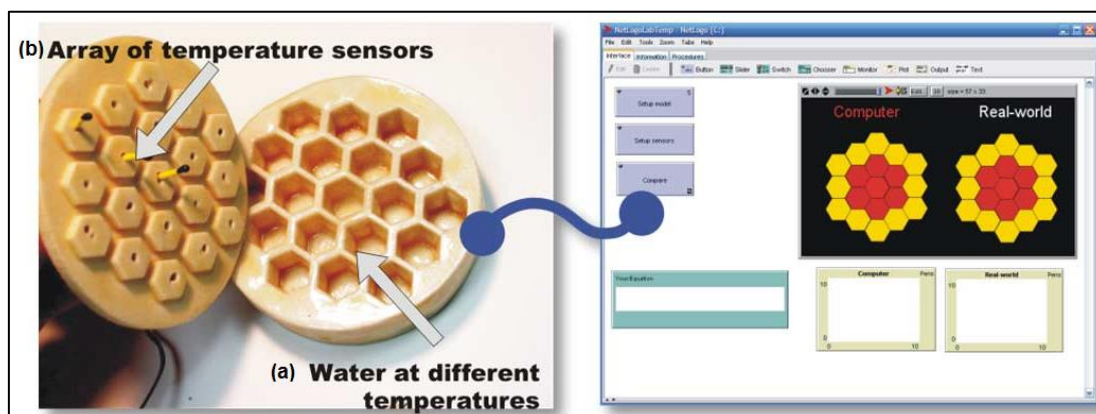
Durante a fase de projeto, os objetivos da modelagem devem ser definidos e as variáveis de interesse devem ser identificadas. Também devem ser selecionados os equipamentos e materiais necessários para a criação do modelo físico, assim como os sensores necessários para a coleta de dados. Para isso, muitas vezes é necessário que seja feita uma busca na literatura, de modo que essas informações forneçam a base para a criação do modelo. (BLIKSTEIN, 2012, p. 3).

Já na fase de construção, o modelo físico deve ser criado e os sensores para coleta de dados devem ser acoplados. Concomitantemente a isso, o modelo computacional deve ser desenvolvido de forma que, se possível, receba e mostre os dados do modelo físico através de uma interface que processe os dados dos sensores. (BLIKSTEIN, 2012, p. 3).

Então, na última fase, ocorre a interação do indivíduo ou grupo com os modelos criados. Nesta fase, deve ser verificado se há convergência entre os dados obtidos durante a simulação dos modelos, comparando-os e analisando-os estatisticamente. Em caso de similaridade no comportamento dos modelos e de correlação entre os dados obtidos, pode-se dizer que o modelo computacional criado é válido. (BLIKSTEIN, 2012, p. 3).

Como exemplo da aplicação da técnica de Modelagem Bifocal, Blikstein (2012, p. 2) mostra um modelo de transferência de calor, que pode ser visto na Figura 8, composto por diversos recipientes organizados de forma hexagonal e preenchidos com água (a). Também é retratada a matriz de sensores usada para coletar os dados (b), assim como o modelo computacional criado com base no fenômeno estudado, com uma representação visual dos recipientes.

Figura 8 – Exemplo de modelo criado por meio de Modelagem Bifocal



Fonte: Blikstein (2012, p. 2).

A conexão entre os experimentos físicos e os modelos digitais pode ser feita através de plataformas eletrônicas como a *GoGo Board*, mas não se limitando a esta, permitindo que dados coletados por sensores eletrônicos sejam exibidos junto aos dados produzidos durante a simulação do modelo digital e que a comparação entre esses dados seja feita. (BLIKSTEIN; WILENSKY, 2007, p. 4).

Nota-se que a estrutura da Modelagem Bifocal não especifica um *software* a ser usado para a criação dos modelos digitais. Cabe ao estudante ou ao pesquisador selecionar o melhor *software* com base em seus requisitos e necessidades. Como exemplos, Blikstein e Wilensky (2007), Blikstein *et al.* (2012) e Blikstein (2012) utilizaram o *software NetLogo* em suas produções. Já Sgorlon e Cruz (2016) construíram seus modelos usando um *software* conhecido como *Algodo*.

Blikstein e Wilensky (2007) propuseram a utilização da técnica de Modelagem Bifocal por alunos de graduação e pós-graduação. Os participantes desse estudo foram divididos em dois grupos: o primeiro grupo foi encarregado de construir modelos puramente virtuais e o segundo grupo utilizou o método da Modelagem Bifocal de forma completa, construindo modelos virtuais e modelos físicos.

Conforme foi relatado na pesquisa, os estudantes que realizaram o experimento de maneira completa notaram certas características dos fenômenos estudados, como perda de energia, reversibilidade, sincronicidade e precisão, que não foram comentadas pelo grupo que construiu apenas modelos virtuais. (BLIKSTEIN; WILENSKY, 2007, p. 8).

Isso evidencia a importância da interação e da comparação entre o modelo físico e o virtual. O indivíduo ou grupo envolvido na tarefa de modelagem, ao comparar os resultados de seus experimentos, pode refletir sobre as características do modelo criado e principalmente sobre as características do próprio fenômeno modelado, raciocinando sobre o que é razoável representar em seus modelos. (BLIKSTEIN; WILENSKY, 2007, p. 10).

Blikstein (2012) concluiu, ao realizar uma pesquisa envolvendo estudantes de nível equivalente aos de ensino médio, que a aplicação da Modelagem Bifocal promoveu um ganho de aprendizado com relação aos fenômenos físicos estudados,

destacando que o fator marcante para isso foi a interação com a construção de modelos.

Uma dificuldade recorrente encontrada na aplicação da Modelagem Bifocal é fazer com que os experimentos físicos e computacionais tenham sincronia, ou seja, fazer com que fenômenos análogos ocorram da mesma forma e ao mesmo tempo tanto no mundo real quanto no mundo digital.

Blikstein (2012) comenta que no *NetLogo*, *software* escolhido em seu estudo, a passagem de tempo é representada por passos discretos, conhecidos como *ticks*, necessitando que exista uma conversão entre essa unidade discreta de tempo e a unidade de tempo contínua na qual os eventos acontecem do mundo real. Nesse caso, a conversão deve ser feita em cada modelo e demanda uma análise da escala de tempo em que o fenômeno modelado ocorre.

Existe também a preocupação com a escala de espaço escolhida. Um modelo deve representar um sistema de forma a incluir as características de interesse e os demais elementos que tenham influência sobre o fenômeno. Logo, a definição da escala de espaço depende fundamentalmente do espaço ocupado pelo fenômeno, processo ou sistema no mundo real. (BLIKSTEIN, 2007).

Ainda assim, com a capacidade de processamento atual, em diversos casos o fenômeno estudado é muito complexo e a escala de espaço, relativamente ao fenômeno fundamental, é muito grande para que sua simulação computacional ocorra em tempo real. Blikstein (2007, p. 9) exemplifica isso comentando a modelagem hipotética de todos os átomos em uma gota de água, algo complexo pela quantidade de variáveis e interações simultâneas e que demandaria um altíssimo poder de processamento.

Em contrapartida a essas dificuldades, a natureza da Modelagem Bifocal determina que os modelos criados sejam validados, contribuindo para a confiabilidade e utilidade desses modelos, e permitindo que sejam usados para a tomada de decisões em diversas áreas.

Logo, através da aplicação dos conceitos de Modelagem Bifocal aos fenômenos de estudo da Engenharia Mecânica espera-se um ganho de aprendizado similar ao observado por Blikstein e Wilensky (2007), Blikstein (2012) e Sgorlon

(2016), permitindo que os estudantes e os profissionais dessa área compreendam mais profundamente os seus objetos de estudo através da união da teoria e da prática.

Assim sendo, o capítulo seguinte apresenta algumas das ferramentas de modelagem disponíveis e que podem ser integradas ao conceito de Modelagem Bifocal, além de ser feita a avaliação do *software NetLogo* como ferramenta de modelagem.

2.5 FERRAMENTAS DE MODELAGEM E SIMULAÇÃO COMPUTACIONAL

Tão importante quanto compreender os fundamentos da Modelagem Bifocal é realizar a escolha correta da ferramenta de modelagem computacional, já que cada uma delas se adequa de maneiras diferentes à modelagem de determinados fenômenos físicos.

Este capítulo procura abordar brevemente algumas dessas ferramentas, assim como busca verificar a validade do *NetLogo* como plataforma de modelagem computacional na área de Engenharia Mecânica. O Quadro 2 aborda algumas das ferramentas de modelagem computacional mais usadas, incluindo suas principais funcionalidades.

Quadro 2 – Ferramentas de modelagem computacional

Ferramenta	Desenvolvedor	Funcionalidades	Abstração
Linguagens de programação de alto nível (C++, Swift, Java, Python)	Diversos	Uso geral	Alto nível
SIMLIB/C++	Faculty of Information Technology	Auxilia na criação de modelos em C++	Alto nível
Arena Simulator	Rockwell Automation	Modelagem por blocos e elementos; Modelagem discreta	Alto nível
MATLAB	MathWorks	Modelagem de sistemas dinâmicos; Análise numérica; Inteligência artificial	Alto nível
Scilab	Scilab Enterprises	Modelagem de sistemas dinâmicos; Análise numérica;	Alto nível

Fonte: Peringer (2017); Vieira et al. (2014); MathWorks (2018); Scilab (2017).

Porém, o foco deste estudo está na criação de modelos baseados em agentes. Um dos expoentes nesta área é o *NetLogo*, melhor detalhado na seção a seguir.

2.5.1 NetLogo

O *NetLogo*, atualmente em sua versão 6.0.4, é um *software* de modelagem e simulação computacional multiplataforma, livre e de código aberto que utiliza o paradigma de modelagem baseada em agentes. Este *software* foi criado por Wilensky (1999) para fins educacionais e de pesquisa, fornecendo uma interface e uma linguagem de programação que pode ser utilizada desde a educação básica até os ambientes acadêmicos de graduação e pós-graduação. (TISUE; WILENSKY, 2004, p. 1).

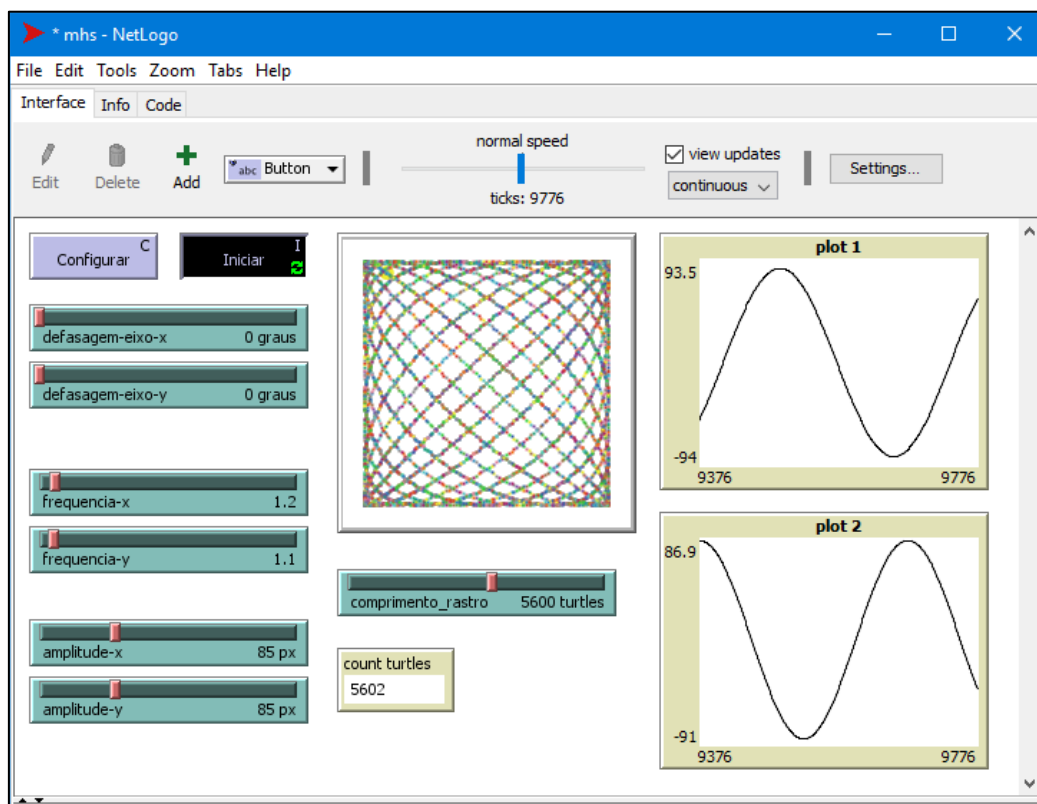
Nesta seção são apresentados alguns dos principais conceitos relacionados ao *NetLogo*, como a interface dessa aplicação, os tipos de agentes usados, a configuração do ambiente de simulação e aspectos de programação.

2.5.1.1 Interface

Em todas as plataformas suportadas, a interface do *NetLogo* consiste de três abas principais, que são *Interface*, *Info* e *Code*, ou Interface, Informação e Código. Na aba interface podem ser adicionados os elementos de controle, como botões, “deslizadores” ou “*sliders*”, “selecionadores” ou “*choosers*”, comutadores ou “*switchs*” e campos para entrada de texto ou “*input*”, assim como podem ser adicionados elementos de monitoramento, como monitores de dados, “*plot*” ou geradores de gráficos e um campo de saída de dados ou “*output*”. (WILENSKY, 2018, p. 101).

A Figura 9 mostra a interface de usuário do *NetLogo*, com vários dos elementos de entrada e saída comentados, como botões, deslizadores, contadores e gráficos. O exemplo mostrado na figura corresponde à simulação da trajetória de uma partícula que se comporta segundo um Movimento Harmônico Simples (MHS) e permite a alteração de diversos parâmetros como frequência de oscilação, amplitude e defasagem.

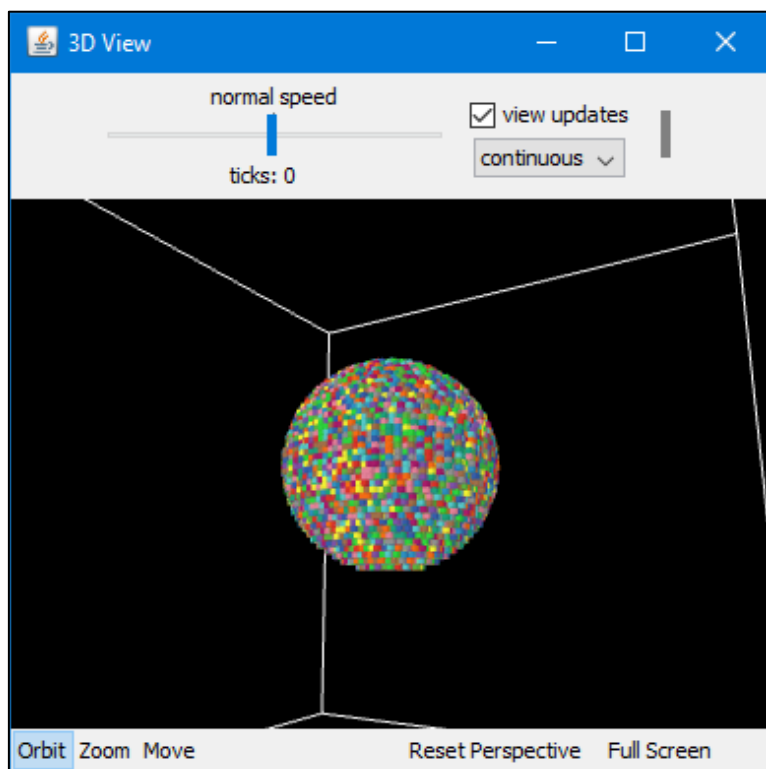
Figura 9 – Interface do NetLogo



Além dos elementos de controle e monitoramento mencionados anteriormente, o *NetLogo* possui um ambiente de simulação conhecido como “mundo” ou “*world*”, no qual habitam os agentes. Este mundo consiste de um retângulo, no caso de um modelo de duas dimensões, ou de um paralelepípedo retângulo, no caso de um modelo criado em três dimensões. (WILENSKY, 2018, p. 102 e 231).

A Figura 10 apresenta uma simulação de exemplo criada para o ambiente tridimensional do *NetLogo*, onde diversos agentes foram organizados de maneira a formar uma esfera.

Figura 10 – Ambiente de simulação do *NetLogo* 3D



O tamanho e as configurações de fronteira desse ambiente de simulação podem ser definidos por meio da interface do *NetLogo* ou através de código, da mesma forma que o local do ponto de origem, ou seja, o ponto de coordenadas (0, 0, 0) em um sistema de coordenadas cartesianas com eixos x, y e z. (WILENSKY, 2018, p. 102).

Já as fronteiras desse ambiente de simulação podem ser fechadas, de forma que os agentes não possam atravessá-las, ou abertas, de maneira que um agente que atravessasse os limites do ambiente, em qualquer das fronteiras, é transportado para a fronteira oposta à que atravessou. (WILENSKY, 2018, p. 152).

2.5.1.2 *Aba de informações*

O *NetLogo* possui uma seção conhecida como “*Info*”, onde podem ser inseridas as informações relevantes sobre o modelo. Essas informações podem conter uma introdução a respeito do fenômeno modelado, os objetivos do modelo e o modo de uso, assim como pode informar sobre características a explorar no modelo. (WILENSKY, 2018).

2.5.1.3 Agentes

Por ser um ambiente de modelagem baseado em agentes, o *NetLogo* incorpora muitos dos conceitos apresentados na seção 2.3, como a capacidade de percepção dos agentes e a interação entre agentes e ambientes. Porém, a linguagem de programação utilizada para criar modelos no *NetLogo* leva em conta diferentes tipos de agentes, conhecidos como *turtles*, *patches*, *links* e *observer*, assim como o uso de uma grande quantidade desses agentes em uma mesma simulação. É importante conceituar estes tipos de agentes para então prosseguir para as outras características do *NetLogo*. (WILENSKY, 2018, p. 122).

2.5.1.4 Turtles

Os agentes do tipo *turtle* possuem uma grande flexibilidade quanto a apresentação gráfica, movimentação e interação. Cada um dos agentes que pertence a esse grupo possui variáveis manipuláveis que definem sua cor e sua forma, assim como posição e orientação em relação a um sistema de coordenadas de referência. Esses agentes também têm capacidade de percepção, podendo identificar ou perceber outros elementos presentes no ambiente, assim como a capacidade de interagir com esses elementos e com o próprio ambiente. (WILENSKY, 2018, p. 122).

Além disso, novas variáveis podem ser adicionadas a cada agente do tipo *turtle* conforme seja necessário representar características específicas de um sistema, como temperatura ou velocidade. Isso pode ser definido por meio do comando “*turtles-own*”, no ambiente de programação, e amplia consideravelmente a capacidade de representar fenômenos físicos diversos através dos agentes disponíveis no *NetLogo*. (WILENSKY, 2018, p. 124).

A Figura 11 mostra como os agentes do tipo *turtle* podem assumir diversas cores, formas, orientações e ainda podem ser dispostos espacialmente conforme desejado. Neste caso, um código simples foi escrito no *NetLogo* com o objetivo de preencher cada elemento do ambiente com agentes do tipo *turtle* de cores, formas e orientações aleatórias.

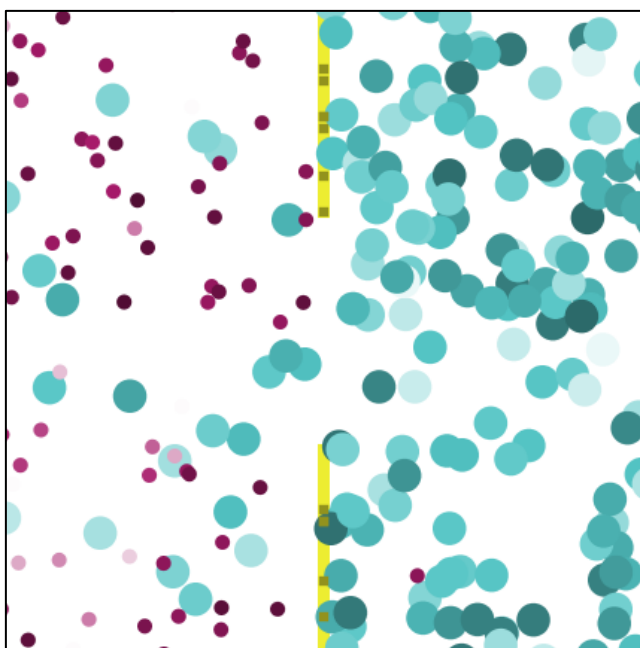
Figura 11 – *Turtles* com formas, cores e orientações aleatórias



Turtles também podem ser subdivididos sob a forma de “espécies”, ou “*breeds*”, fazendo com que diferentes grupos possam ser representados em uma mesma simulação. Cada um dos elementos dessas espécies pode receber um conjunto de variáveis adicionais, tal como massa ou tamanho, por exemplo, dando a cada grupo características e comportamentos distintos. (WILENSKY, 2018, p. 131).

Wilensky (1997) demonstra isto por meio de modelo da interação entre dois gases com propriedades físicas distintas e confinados em uma caixa, que pode ser visto na Figura 12, onde as partículas de ambos são representadas por agentes de espécie (*breed*) “particles”, de formato circular e de cores magenta e ciano. Quando há colisão das partículas com as paredes da caixa são criados agentes da espécie “flashes”, indicando por um breve período os locais onde houve essa colisão.

Figura 12 – Modelo de interação entre dois gases



Fonte: Adaptado de Wilensky, 1997.

2.5.1.5 *Links*

Já os agentes do tipo *link*, ou agentes de ligação, possuem a função de conectar dois outros agentes do tipo *turtle*. Essa ligação é representada visualmente como uma linha que vai de um agente ao outro e só existe enquanto os dois agentes existirem na simulação. (WILENSKY, 2018, p. 131).

2.5.1.6 *Patches*



Além disso existem os *patches*, que são agentes sem capacidade de movimentação, fazendo parte do ambiente de simulação. Ainda assim, esses agentes possuem certas variáveis que definem suas características, como *pcolor*, que define sua cor, e podem receber variáveis personalizadas através do comando *patches-own*. (WILENSKY, 2018, p. 122).

A linguagem de programação do *NetLogo* possibilita definir as cores dos *patches* com base em uma imagem digital, sendo que cada agente do tipo *patch* pode receber a cor correspondente do *pixel*³ ou do conjunto de *pixels* que possuem coordenadas aproximadas às do *patch*. Dessa maneira, ambientes inteiros podem ser importados de maneira rápida, necessitando apenas de uma linha de código. (WILENSKY, 2018, p. 392).

Para que isso possa ser realizado, a imagem digital deve estar gravada no mesmo diretório em que está o modelo do *NetLogo*. É o que mostra o item (a) da Figura 13. Já o item (b) apresenta o comando utilizado para a importação dessa imagem como ambiente do *NetLogo*. Por fim, os *patches* assumem as cores da imagem importada (c), e o ambiente, composto pelo agregado de *patches*, assume a forma dessa mesma imagem.

³ *Picture Element*, ou Elemento de Imagem

Figura 13 – Importação de uma imagem como ambiente do *NetLogo*

 <p>Imagem.png Modelo.nlogo</p> <p>(a) Imagem digital junto ao modelo.</p>	<pre>import-pcolors "Imagem.png"</pre> <p>(b) Comando de importação.</p>	 <p>(c) <i>Patches</i> assumem as cores e o formato da imagem.</p>
--	--	---

2.5.1.7 Observer

O observador, ou *observer* no *NetLogo*, é um agente que não possui localização. Esse agente, como sugere seu nome, pode observar todos os outros agentes, porém sua função principal é instruir os outros agentes de acordo com os comandos fornecidos na programação. (WILENSKY, 2018, p. 122).

2.5.1.8 Linguagem de Programação

O *NetLogo* possui um enorme vocabulário de termos relacionados a programação e que podem ser vistos mais detalhadamente em Wilensky (2018). No entanto, apenas alguns dos itens mais importantes e relevantes para este trabalho são destacados, como primitivas, procedimentos e os tipos de variáveis disponíveis.

As primitivas, ou *primitives*, são funções nativas do *NetLogo*, já embutidas em sua linguagem de programação. Exemplos de primitivas são *create*, que cria um agente do tipo *turtle*, e *die*, que elimina os agentes escolhidos. (WILENSKY, 2018, p. 122).

Além das primitivas, a linguagem de programação do *NetLogo* suporta estruturas de programação conhecidas como *procedures*, ou procedimentos, que são semelhantes às funções, presentes em outras linguagens de programação, como C++. Esses procedimentos podem ser executados a partir de elementos de interface, por outros procedimentos ou mesmo pelo observador, e incluem a capacidade de

receber parâmetros externos no momento de sua invocação. (WILENSKY, 2018, p. 122 e 123).

Um procedimento deve ser iniciado com o comando *to*, seguido do nome desse procedimento. Após isso os comandos relacionados a esse procedimento podem ser adicionados, como “*fd X*” para mover um agente *turtle* em X unidades ou “*set color Y*” para alterar a cor desse agente para uma cor Y, definida numericamente. (WILENSKY, 2018, p. 122).

Wilensky (2018, p. 123) diz que muitos modelos do *NetLogo* possuem um botão que executa um procedimento de configuração, normalmente nomeado “*setup*”. Dentro desse procedimento podem ser adicionados todos os comandos relativos à inicialização do modelo, como o comando que reseta o contador de tempo da simulação (*reset-ticks*) e o comando que remove todos os agentes criados em uma simulação anterior (*clear-all*).

Além disso, é importante que seja criado um procedimento de execução, associado a um botão e muitas vezes denominado “*go*”, configurado para que seja executado continuamente, como em um laço de repetição, até que o botão de execução seja pressionado novamente. Neste procedimento podem ser adicionados diversos comandos relativos ao modelo, incluindo criação e manipulação de agentes e incremento do tempo de simulação. (WILENSKY, 2018, p. 123).

Também pode ser adicionada a um procedimento a estrutura condicional “*if*”, que executa uma ação com base na validade de uma condição lógica, assim como a estrutura de repetição “*while*”, que executa repetidas vezes o mesmo conjunto de comandos até que uma condição lógica se torne falsa. (WILENSKY, 2018, p. 391 e 432).

A linguagem de programação do *NetLogo* suporta diversos tipos de variáveis que abrangem elementos distintos. O primeiro desses tipos de variáveis são as variáveis globais, ou *globals*, que estão acessíveis a todos os agentes do modelo. Essas variáveis podem ser criadas por meio de código ou ao adicionar elementos de interface como *sliders* ou *choosers*. Variáveis desse tipo podem representar características globais como temperatura inicial ou aceleração gravitacional. (WILENSKY, 2018, p. 124).

Existem conjuntos de variáveis específicas a cada um dos tipos de agentes, que são as variáveis de *turtles*, definidas pelo comando “*turtles-own*”, as variáveis de *patches*, definidas pelo comando “*patches-own*”, e as variáveis de links, definidas pelo comando “*links-own*”. Cada espécie de agentes também pode receber um conjunto de variáveis personalizadas por meio do comando “<espécie>-own”. (WILENSKY, 2018, p. 124).

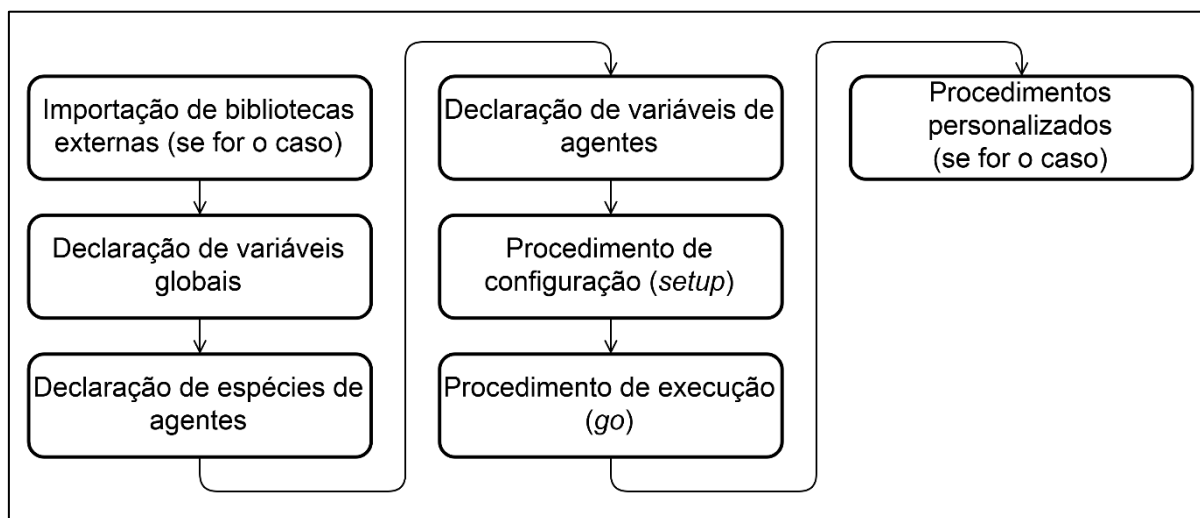
Cada agente individual, de qualquer um dos tipos, pode receber valores diferentes para a mesma variável, como velocidade, cor ou temperatura do agente, e cada agente pode requisitar a outro agente o valor de suas variáveis. (WILENSKY, 2018, p. 124).

O tempo, em uma simulação do *NetLogo*, passa em intervalos discretos, conhecidos como *ticks*. Por padrão, essa escala de tempo é incrementada por uma unidade quando o comando o comando “*tick*” é executado. Porém, pode-se definir o valor desse incremento em código por meio do comando “*tick-advance*”. Outra função do comando “*tick*” é atualizar a visualização do mundo de simulação, permitindo que as mudanças de estado e do comportamento geral dos agentes sejam visualizadas na tela em tempo real. (WILENSKY, 2018, p. 125).

Como forma de promover integração entre *hardware* e *software* e para incrementar a funcionalidade dos modelos, o *NetLogo* oferece suporte a bibliotecas e recursos externos sob a forma de extensões. Algumas dessas extensões incluem funcionalidades como comunicação com a plataforma *Arduino*, comunicação com a plataforma *GoGo*, manipulação de matrizes (*arrays*), utilização de um maior espaço de cores e a capacidade de realizar operações de computação estatística por meio da linguagem *R*. (WILENSKY, 2018, p. 248-325).

A estrutura básica do código de um modelo no *NetLogo* é representada através de fluxograma na Figura 14, com importação de bibliotecas externas, declaração de variáveis globais, declaração de espécies de agentes, declaração de variáveis de agentes, criação de um procedimento de configuração (*setup*) e criação de um procedimento de execução (*go*). Além desses elementos, novos procedimentos podem ser adicionados conforme sejam necessários. (WILENSKY, 2018).

Figura 14 – Estrutura básica do código de um modelo do *NetLogo*



Como pode ser percebido, apenas os conceitos e elementos mais importantes relacionados à linguagem de programação do *NetLogo* e que serão utilizados neste trabalho foram abordados. Existe uma infinidade de ferramentas e elementos disponibilizados para utilização no *NetLogo*, logo, recomenda-se a consulta ao Manual de Usuário do *NetLogo*, disponível apenas na língua inglesa em Wilensky (2018), para mais informações.

2.5.1.9 Espaço de Comportamentos

O *NetLogo* oferece um módulo conhecido como *BehaviorSpace*, ou Espaço de Comportamentos, que permite que a simulação de um modelo seja executada tantas vezes quanto for necessário, de maneira automatizada e com variação de parâmetros de simulação entre cada execução. Isso permite que o espaço de possibilidades de comportamento do modelo seja explorado, fornecendo uma percepção sobre o comportamento do modelo em diversas condições de operação. Unindo essa funcionalidade à capacidade de computação estatística do *NetLogo* é possível realizar análises quantitativas utilizando os dados dos modelos criados. (WILENSKY, 2018).

2.5.2 Validade do *NetLogo* como ferramenta de modelagem computacional

Como foi comentado anteriormente, cada *software* de modelagem computacional se adequa melhor a determinados contextos. Logo, é importante avaliar a validade do *NetLogo* no contexto e nas necessidades da Engenharia

Mecânica para que possa ser usado como plataforma de modelagem computacional neste trabalho.

Para isso, é preciso identificar alguns requisitos desejados de um *software* de modelagem computacional. Law e Kelton (2000, p. 208, tradução própria) enunciam como requisitos os seguintes itens:

1. Capacidades gerais (incluindo facilidade de modelagem e facilidade de uso);
2. Considerações de *hardware* e de *software*;
3. Animação;
4. Recursos de estatística;
5. Suporte ao cliente e documentação;
6. Relatórios de saída e gráficos.

Com isso em mente, nos parágrafos seguintes o *NetLogo* será avaliado em relação a cada um dos itens citados.

Law e Kelton (2000, p. 208) consideram que uma das características mais importantes de um *software* de modelagem computacional é o que chamam de “flexibilidade de modelagem”, ou seja, a capacidade de permitir a criação de modelos que possuam uma quantidade qualquer de complexibilidade.

Em face a isso, durante a concepção do *NetLogo*, conforme relatado em Tisue e Wilensky (2004), a ideia geral foi criar um ambiente de modelagem com capacidade de modelar sistemas complexos e com comportamentos que evoluem com o tempo. Para isso, o *NetLogo* utiliza agentes genéricos e que devem ser programados de acordo com a necessidade do fenômeno de estudo.

Conforme foi comentado anteriormente na seção 2.5.1.8, a linguagem de programação do *NetLogo* permite a criação de variáveis de uso global e de variáveis específicas de agentes, incluindo a capacidade de usar essas variáveis em operações lógicas, incrementando assim a gama e a complexibilidade de sistemas possíveis de serem modelados. (WILENSKY, 2018, p. 124).

Além disso, a interface de usuário do *NetLogo* possui elementos de monitoramento de agentes que permitem acompanhar um ou mais agentes durante a

simulação, permitindo também que suas variáveis sejam monitoradas e alteradas dinamicamente, facilitando a usabilidade e o processo de depuração de erros em um modelo. (WILENSKY, 2018, p. 77).

Logo, com essas informações é possível concluir que esse *software* possui grande versatilidade e capacidade de adaptação, atendendo ao primeiro dos requisitos estipulados por Law e Kelton (2000, p. 208).

Já em relação ao segundo requisito estipulado por Law e Kelton (2000, p. 208), o Manual de Usuário do *NetLogo* (WILENSKY, 2018, p. 55) informa como requisitos de *software* que a versão 6.0.3 do *NetLogo* pode ser executada nos sistemas operacionais *Windows 10*, *Windows 8*, *Windows 7* e *Windows Vista*, assim como nas versões do *macOS*⁴ superiores à versão 10.8.3 e também em ambientes *Linux* baseados em *Debian* e *Red Hat*. Em outras palavras, o *NetLogo* pode ser executado em todos os principais sistemas operacionais atuais.

Dessa maneira, computadores e dispositivos de *hardware* capazes de executar qualquer um dos sistemas operacionais mencionados, tanto nas versões de 32 bits quanto nas versões de 64 bits, também pode executar o *NetLogo*, com desempenho variável em relação ao poder computacional do equipamento. (WILENSKY, 2018, p. 55).

No entanto, Wilensky (2018, p. 359) informa que o *NetLogo*, em sua configuração atual, não pode ser executado em dispositivos móveis que operam com os sistemas operacionais *Android* ou *iOS*.

Assim, entende-se que o *NetLogo* pode ser executado nativamente na maior parte dos computadores disponíveis nas instituições de ensino e nos laboratórios de pesquisa, tanto em sistemas operacionais gratuitos e de código aberto quanto em sistemas operacionais pagos e de código fechado, sendo uma ferramenta acessível tanto em custo quanto em seus requisitos de *hardware* e *software*.

Prosseguindo ao terceiro requisito, Law e Kelton (2000, p. 210, tradução própria) dizem que “a disponibilidade integrada de animações é uma das razões para o aumento no uso de modelagem para simulação.” A partir desse pensamento, é notável a maneira visual com que o *NetLogo* apresenta o ambiente de simulação dos

⁴ Anteriormente conhecido como *Mac OS X*.

modelos, mostrando as mudanças de estado de elementos individuais e do agregado de agentes de forma visível e interativa (WILENSKY, 2018).

Então, identifica-se que o *NetLogo* possui a capacidade de exibir o estado da simulação dos modelos de maneira dinâmica, como requisitado por Law e Kelton (2000, p. 208).

O quarto requisito de *software* estipulado por Law e Kelton (2000, p. 208) tem relação com a capacidade de um *software* de modelagem e simulação computacional realizar operações de estatística que apresentem resultados e informações sobre um determinado modelo.

Sobre isso, pode-se afirmar que o *NetLogo* disponibiliza meios para a realização de operações estatísticas por intermédio da extensão R, invocada pelo comando “*extensions [r]*”, que possui diversas ferramentas e comandos que auxiliam na extração e processamento de dados do modelo computacional. (WILENSKY, 2018, p. 325). Logo, o *NetLogo* também atende ao requisito de processamento estatístico.

Grande parte das informações disponibilizadas sobre o *NetLogo* e apresentadas neste trabalho foram retiradas do Manual de Usuário do *NetLogo* (WILENSKY, 2018), que possui desde um histórico das mudanças realizadas nesse *software* ao longo de todas as suas versões até uma explicação sobre a função e o uso das diversas primitivas e funções presentes em seu código e disponíveis ao usuário.

Junto ao *NetLogo* é fornecida uma biblioteca de código aberto contendo uma grande quantidade de modelos de exemplo que têm relação com diversas áreas do conhecimento e auxiliam o usuário a interagir, buscar recursos e compreender o funcionamento e as possibilidades dessa aplicação, conforme informado por Wilensky (2018, p. 359).

Além disso, na literatura existem obras como a de Wilensky e Rand (2015), que discorre sobre Modelagem Baseada em Agentes com aplicações de modelos de exemplo no *NetLogo*, e diversos artigos que apresentam o funcionamento, a filosofia de criação e aplicações de uso do *NetLogo*, como pode ser visto em Tisue e Wilensky (2004) e Blikstein *et al* (2012), entre outros.

No entanto, o mesmo Manual de Usuário do *NetLogo* (WILENSKY, 2018) informa que essa aplicação é distribuída ao público geral sem garantia de resultados,

sendo que cabe ao usuário verificar a validade dos modelos criados. Ainda assim existem licenças comerciais do *NetLogo* que podem ser adquiridas entrando em contato diretamente com o criador do *software*.

Entende-se, então, que o *NetLogo* possui vasta documentação e recursos que auxiliam o usuário na tarefa de criar modelos computacionais, inclusive com a possibilidade de que uma licença comercial de *software* seja adquirida, caso seja necessário, atendendo ao quinto requisito de *software* de Law e Kelton (2000).

Já em relação ao último requisito de *software* listado pelo autor, é possível afirmar que no *NetLogo* estão disponíveis gráficos que são atualizados em tempo real e apresentam a variação de uma ou mais variáveis do modelo em relação ao tempo de simulação. Da mesma maneira, o *NetLogo* suporta comandos de saída de dados, seja para mostrá-los na tela ou para gravá-los em arquivos para análise posterior. (WILENSKY, 2018, p. 144 e 148).

Então, a partir das informações apresentadas, é possível afirmar que o *NetLogo* atende de maneira satisfatória aos requisitos identificados por Law e Kelton (2000), se mostrando uma ferramenta válida para a modelagem de fenômenos físicos. Ainda assim, é importante verificar as aplicações do *NetLogo* como ferramenta de modelagem computacional para fenômenos físicos estudados na Engenharia Mecânica.

É possível verificar que a Biblioteca de Modelos do *NetLogo* (WILENSKY, 1999) contém modelos computacionais de diversas áreas, como física, química, ciência de materiais, termodinâmica, matemática, psicologia e ciências sociais, entre outras áreas. Uma pesquisa nesta biblioteca revela que alguns modelos de interesse para a Engenharia Mecânica estão disponíveis para consulta e modificação, como modelos de difusão térmica, de turbulência em fluidos, do comportamento de gases, da dinâmica de polímeros e do crescimento dos grãos de diferentes materiais. (WILENSKY, 1997, 1998, 2003, 2005; BLIKSTEIN; WILENSKY, 2005).

Nota-se, porém, que o modelo de transferência térmica mencionado apresenta limitações quanto ao formato e resolução da região onde ocorrem as trocas térmicas. O modelo de condução de calor desenvolvido no presente trabalho, de maneira independente ao modelo de Wilensky (1998), busca proporcionar maior flexibilidade e personalização ao usuário, permitindo a importação de diferentes formatos de amostra

e maior interação durante a simulação por meio da possibilidade de inclusão de novas fontes de calor dinamicamente.

Assim, entende-se que há precedentes para a modelagem de fenômenos físicos estudados na graduação em Engenharia Mecânica no *NetLogo*, corroborando para a validade da escolha do *NetLogo* como plataforma de modelagem neste trabalho.

2.6 CONDUÇÃO TÉRMICA

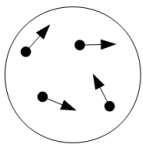
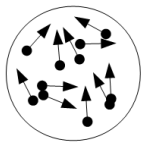
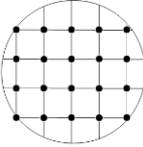
Para que a modelagem de um sistema tenha êxito é fundamental buscar na teoria os conceitos associados aos fenômenos físicos de interesse. Desse modo, nesta seção são apresentados os fundamentos teóricos relacionados ao fenômeno físico de condução térmica.

Contida no campo de estudo dos Fenômenos de Transporte, a transmissão de calor pode ser conceituada como “[...] a energia em trânsito devido a uma diferença de temperatura.” Essa transmissão de calor pode ocorrer na forma de condução, em sólidos e em fluidos estáticos, de convecção, em fluidos que se movem, e de radiação, no caso da energia transferida por meio de ondas eletromagnéticas. (INCROPERA; DEWITT, 1998, p. 2).

Incropera e DeWitt (1998, p. 2) dizem que o fenômeno físico elementar da transmissão de calor por condução ocorre a níveis atômicos e moleculares, e pode ser identificado pela “[...] transferência de energia de partículas mais energéticas para partículas de menor energia, em um meio devido a interações que existem entre elas.”

Logo, em seu nível mais elementar, o fenômeno de condução de calor trata da transferência de energia de um átomo a outro, por meio da migração de elétrons e por meio de ondas vibracionais na estrutura dos átomos, como pode ser observado na Figura 15. Isso ocorre sempre do ponto que possui mais energia térmica, ou maior temperatura, para o ponto menos energético, de menor temperatura. (INCROPERA; DEWITT, 1998, p. 2 e 26).

Figura 15 – Condução de calor em gases, líquidos e sólidos

 <p>Colisões moleculares; Difusão molecular.</p> <p>(a) Gases</p>	 <p>Colisões moleculares; Difusão molecular.</p> <p>(b) Líquidos</p>	 <p>Vibrações da rede; Fluxo de elétrons livres.</p> <p>(c) Sólidos</p>
--	---	--

Fonte: Adaptado de Çengel e Ghajar (2012, p. 22)

A equação que descreve o fluxo de condução de calor unidimensional em um material rígido ou em um líquido estático é conhecida como Lei de Fourier, apresentada a seguir, na equação 1, sob a forma diferencial, onde o termo k [W/m. K] é a constante de condutividade térmica, precedida por um sinal negativo que deriva da propriedade de que o calor é transmitido no sentido da diminuição da temperatura. Já o termo dT/dx tem relação com a variação infinitesimal de temperatura ao longo do material. (FOURIER, 1822 *apud* INCROPERA; DEWITT, 1998, p. 3).

$$q_x'' = -k \frac{dT}{dx} \quad (1)$$

A constante de condutividade térmica k é uma propriedade de cada material, que varia com a temperatura, e está relacionada à capacidade de um material de conduzir calor. Materiais com valores de condutividade térmica altos, como ferro e cobre, são considerados bons condutores de energia térmica. Já materiais com baixos valores de condutividade térmica, como borracha e fibra de vidro, são considerados isolantes de energia térmica. (ÇENGEL; GHAJAR, 2012, p. 20 e 23). Os valores de condutividade térmica de alguns materiais são mostrados no Anexo A, organizados dos materiais mais condutores de energia térmica para os materiais menos condutores de energia térmica.

Porém, segundo definido por Fourier e apresentado por Incropera e DeWitt (1998, p. 3), o fluxo de calor também pode ser definido entre dois pontos de distância L e temperaturas T_1 e T_2 para casos em que a distribuição de calor é linear, simplificando a solução analítica de problemas desse tipo. Assim, a equação 1 pode ser escrita da seguinte forma.

$$q_x'' = k \frac{T_1 - T_2}{L} = k \frac{\Delta T}{L} \quad (2)$$

Onde ΔT é o mesmo que $T_1 - T_2$ e representa a diferença de temperatura entre os pontos onde ocorre a troca térmica.

No entanto, para casos tridimensionais tanto a equação 1 quanto a equação 2 não são satisfatórias. Logo, a Lei de Fourier pode ser escrita na equação 1-3 de forma a considerar, em um sistema de coordenadas cartesianas, a condução térmica ao longo de três dimensões, representadas pelos eixos coordenados x , y e z . (INCROPERA; DEWITT, 1998, p. 25).

$$q'' = -k \nabla T = -k \left(i \frac{\partial T}{\partial x} + j \frac{\partial T}{\partial y} + k \frac{\partial T}{\partial z} \right) \quad (3)$$

Além dessa equação, que descreve o fluxo de calor por condução, Silva Neto, Valda e Lopes (2009, p. 4) citam a equação da difusão de calor bidimensional, apresentada neste trabalho como a Equação 1-4, que trata da análise em regime transiente da distribuição de temperatura em um material. O termo \dot{e} [W/m²] está ligado à geração de energia. Já α [m²/s], é o coeficiente de difusividade térmica do material e pode ser definido como $k/\rho c_p$.

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\dot{e}}{k} = \frac{1}{\alpha} \frac{\partial T}{\partial t} \quad (4)$$

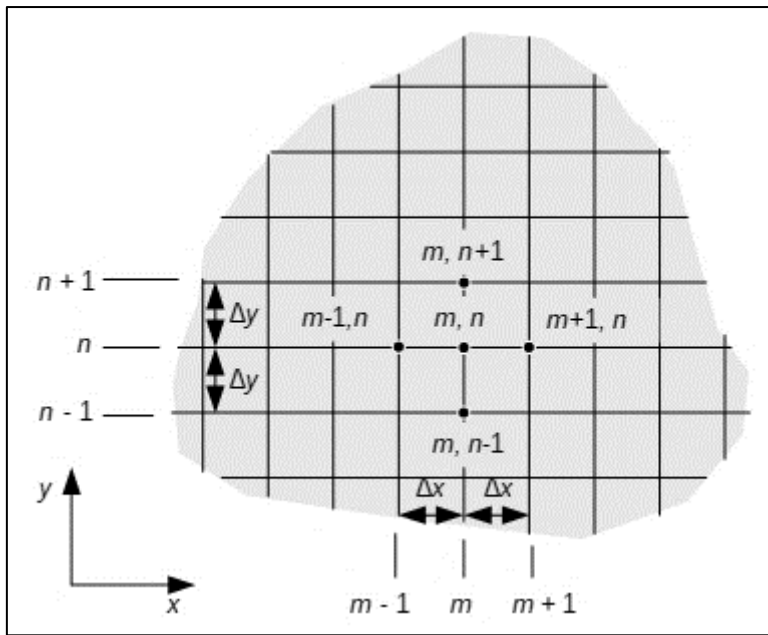
As equações 3 e 4 podem ser solucionadas de maneira a fornecer um resultado analítico para o problema de estudo, porém, neste trabalho pretende-se chegar a soluções por meio de métodos numéricos associados à modelagem baseada em agentes. Maliska (2004, p. 2) informa que a utilização de métodos numéricos para solução de problemas na Engenharia tem a vantagem de fornecer uma solução satisfatória para problemas complexos, onde as simplificações necessárias para que se possa obter uma solução analítica divergem demasiadamente do problema físico real.

Embora não seja o foco principal de sua obra, Maliska (2004, p. 4) cita o método das diferenças finitas como um dos métodos relevantes para a modelagem de fenômenos de transferência de calor. Esse método consiste em discretizar ou dividir

o objeto de estudo em uma malha composta de nós ou pontos com o objetivo de conhecer ou calcular as variáveis em cada um desses pontos.

A Figura 16 mostra a malha utilizada neste método, composta de nós m e n posicionados a uma distância Δx e Δy dos nós mais próximos, $m - 1, n$; $m + 1, n$; $m, n - 1$ e também $m, n + 1$.

Figura 16 – Malha das diferenças finitas



Fonte: Adaptado de Çengel e Ghajar (2012).

Çengel e Ghajar (2012) definem, por meio do método do balanço de energia, que a formulação da equação de transferência de calor permanente por diferenças finitas para um nó interno m, n pode ser representada na equação 5, onde $l = \Delta x = \Delta y$.

$$T_{m-1,n} + T_{m+1,n} + T_{m,n+1} + T_{m,n-1} - 4T_{m,n} + \frac{\dot{e}_{m,n}l^2}{k} = 0 \quad (5)$$

Porém, caso não exista geração de calor no elemento analisado, a equação para um elemento interno se torna $T_{m,n} = (T_{m-1,n} + T_{m+1,n} + T_{m,n+1} + T_{m,n-1})/4$, ou seja, a média aritmética da temperatura dos elementos imediatamente acima, abaixo, à esquerda e à direita.

Dessa maneira, o método das diferenças finitas foi usado neste trabalho para a criação do modelo computacional. Os agentes foram organizados de maneira a

representar os nós da malha e utilizaram a capacidade de percepção para solicitar aos agentes vizinhos a temperatura de cada um deles para assim realizar o cálculo de sua própria temperatura. Da mesma forma, no experimento físico a amostra teve os mesmos pontos delimitados e a temperatura nesses pontos foi aferida por meio de sensores térmicos conhecidos como termopares.

3 DESENVOLVIMENTO

Conforme descrito na seção 2.6, o experimento de condução térmica tem relevância com o curso de Engenharia Mecânica e foi selecionado para a construção de um modelo computacional por meio de Modelagem Bifocal.

É importante notar que a simulação computacional do modelo foi realizada em um computador HP ProBook 4530s, com processador Intel® Core™ i3-2330M de 2.2 GHz, 6 GB de memória RAM DDR3 de 1333 MHz, placa de vídeo integrada Intel® HD Graphics 3000 e tela integrada com resolução de 1366x768. O *software* NetLogo, em sua versão x64, foi executado em ambiente Windows 10 x64. Logo, essas configurações foram levadas em conta para atender aos requisitos de desempenho e corresponder aos fenômenos reais.

O objetivo do experimento de condução térmica é avaliar o perfil de temperatura em uma placa metálica que seja aplicada a fontes térmicas e esteja isolada dos fenômenos de convecção e de radiação, sendo considerado apenas o fenômeno de condução de calor.

Com base nessas informações foi determinado que as variáveis de entrada ou características manipuladas neste experimento são a temperatura ambiente, temperatura da fonte fria e temperatura da fonte quente. Já a variável de saída é a temperatura mensurada em 5 pontos do material.

3.1 EXPERIMENTO FÍSICO DE CONDUÇÃO TÉRMICA

O aparato para o estudo de condução de calor foi construído com chapas de MDF de 3 mm de espessura cortadas a laser no Laboratório de Inovação Tecnológica na Educação (LITE). Nota-se que as dimensões gerais da estrutura montada são de 200 mm de largura, 220 mm de comprimento e 80 mm de altura, conforme os desenhos técnicos do Apêndice D, com 514 g de massa.

Uma das preocupações que surgiram durante o desenvolvimento da estrutura foi que a água proveniente do gelo utilizado como fonte fria pudesse fluir pela amostra e influenciar os resultados, portanto a parte superior do equipamento foi inclinada a um ângulo de 20° em relação à base.

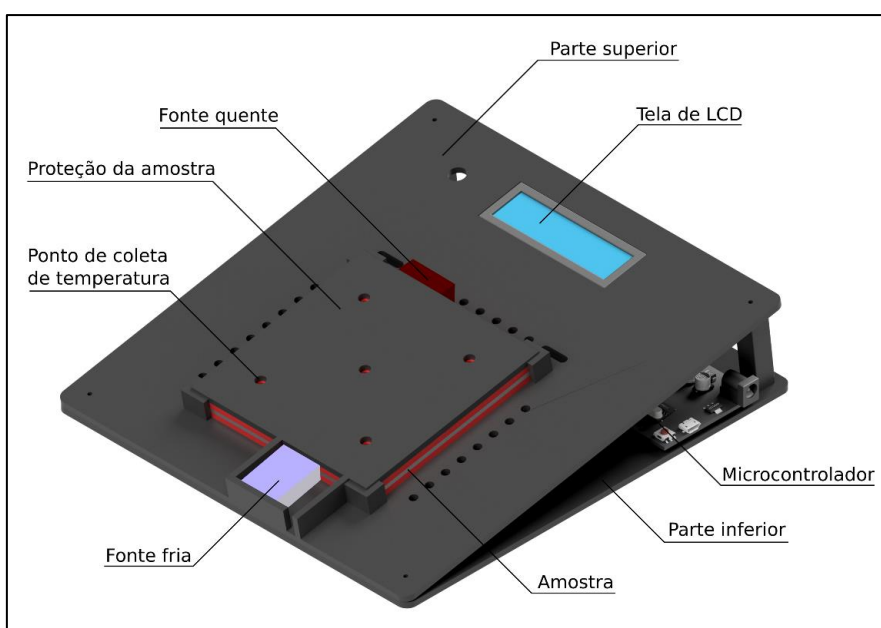
A amostra escolhida para esse estudo é uma placa de aço inoxidável com 100 mm de altura, 100 mm de largura e 0,8 mm de espessura. Conforme disponível em MatWeb (2018), a condutividade térmica média desse material é de 16,2 W/m.K entre 0 °C e 100 °C, que varia conforme sua temperatura, e sua massa específica é de 8 g/cm³, porém não foi possível obter um certificado do material da amostra que corrobore essas informações.

Conforme Thermttest (2018) a condutividade térmica do aço inoxidável à 20 °C é de 14,9 W/m.K, à 50 °C é de 15,7 W/m.K e à 100 °C é de 16,7 W/m.K. Já Sullivan (2018) afirma que à 25 °C essa propriedade é de 16 W/m.K e à 125 °C é de 17,5 W/m.K. Portanto, a condutividade térmica do aço inoxidável aumenta conforme sua temperatura é elevada.

Foi possível determinar que a massa dessa amostra, aferida em laboratório, é de 63 g, logo é possível calcular que a massa específica do material da amostra é de 7,88 g/cm³. Essa amostra foi isolada do ambiente por camadas de espuma de polietileno colocadas entre a placa metálica e as chapas de MDF. Essas espumas de isolamento foram fixadas à estrutura por meio de cola quente.

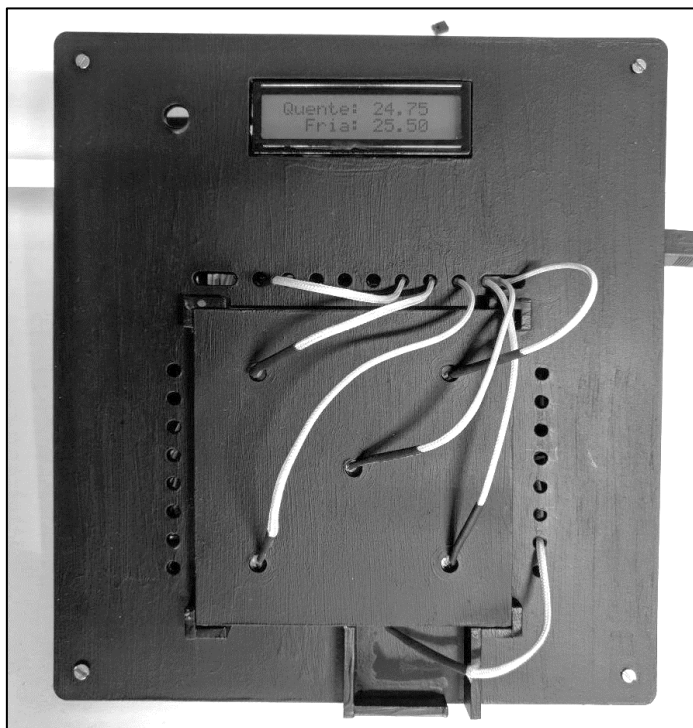
A Figura 17 ilustra o projeto da estrutura do experimento, onde estão destacadas as posições das fontes de calor, dos pontos de coleta de temperatura, e da própria amostra analisada. No entanto, tanto nessa figura quanto na Figura 18 não estão presentes as espumas de isolamento das laterais da amostra.

Figura 17 – Projeto do experimento de condução



Já a Figura 18 mostra a montagem final do experimento. Percebe-se nessa figura que os sensores de temperatura atravessam a parte superior da estrutura e tocam a amostra por meio de furos na proteção da amostra, sendo fixados nessa posição com cola quente.

Figura 18 – Experimento de condução montado



Na primeira etapa do experimento um resistor de 10 W, 1 Ω e 5% de tolerância foi usado como fonte quente, ligado a uma fonte elétrica digital de corrente direta com tensão e corrente variáveis, manipuladas de maneira que a temperatura da superfície do resistor atingisse o valor desejado, transmitindo energia térmica para a amostra.

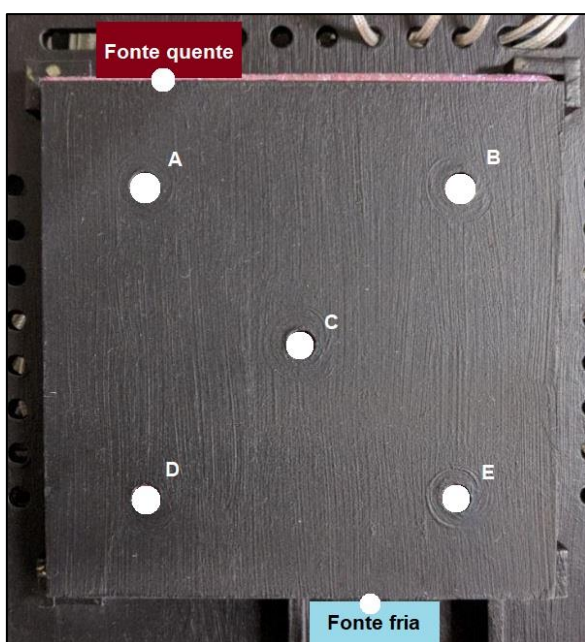
Cubos de gelo apoiados contra a lateral da amostra foram usados como fonte fria de maneira a receber a energia térmica do material. Pequenas molas foram usadas para manter o gelo em contato mesmo após seu derretimento parcial.

No entanto, posteriormente à medida realizada com esses componentes de fonte fria e de fonte quente percebeu-se a necessidade de alterá-los por pastilhas termoelétricas Peltier TEC1-12715 como forma de proporcionar maior estabilidade e confiabilidade às leituras obtidas.

Como sensores de temperatura, foram usados sete termopares do tipo K com extremidade de 1 mm e cabo isolante com comprimento de 1 m, ligados a sete placas MAX 6675 para que os valores analógicos de tensão dos termopares sejam convertidos em sinais digitais, recebidos e processados por um microcontrolador Arduino Uno.

Cinco dos termopares usados no experimento foram ligados diretamente à amostra, enquanto que os dois restantes foram acoplados à fonte quente e à fonte fria. Os termopares responsáveis por coletar a temperatura da amostra foram distribuídos de maneira simétrica, sendo que um termopar foi colocado no centro enquanto os outros quatro termopares foram posicionados a 20 mm das bordas da amostra, como é mostrado na Figura 19.

Figura 19 – Localização dos sensores de temperatura



Também é evidente na Figura 19 que cada ponto de coleta de temperatura recebeu um código, sendo “Quente” o termopar localizado na fonte quente, “Fria”, o termopar localizado junto à fonte fria, A, o termopar localizado próximo ao canto superior esquerdo, B, localizado na porção superior direita, C, ao centro, D, na parte inferior esquerda e, finalmente, E, localizado próximo ao canto inferior direito.

Para criar um perfil de temperatura que proporcione uma grande diferença entre os pontos de coleta, a fonte quente foi posicionada na parte superior da amostra,

próxima à extremidade superior esquerda, oposta à fonte fria, localizada na parte inferior da amostra, próxima ao canto inferior direito.

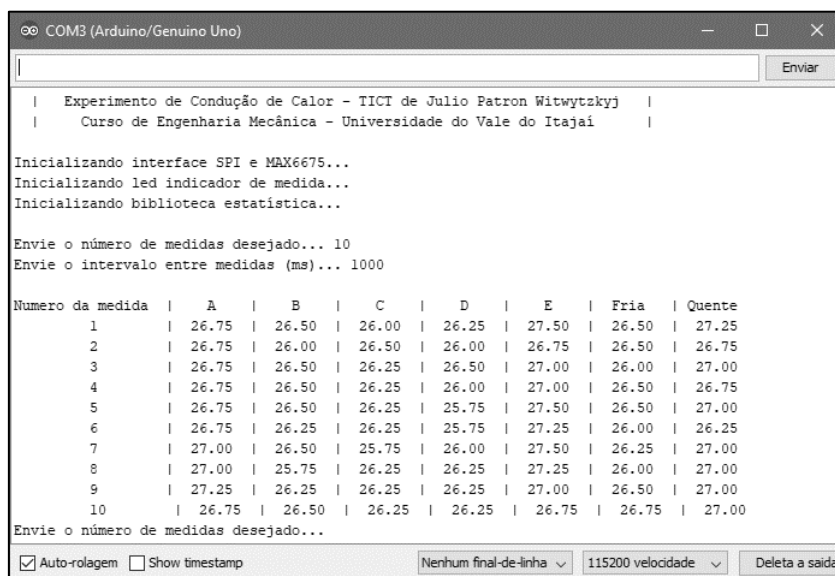
A calibração dos sensores foi feita em temperatura ambiente, 24,57 °C conforme a média de temperaturas aferidas em 7 termômetros analógicos. Foram realizadas 31 medições de cada termopar com 1000 ms de intervalo. Em seguida a média das medidas de cada termopar foi subtraída da temperatura ambiente mensurada.

No momento da coleta de dados percebeu-se a necessidade de aplicar uma pequena quantidade de pasta térmica aos pontos A, B, C, D e E como forma de facilitar a transferência de energia térmica e diminuir a diferença de temperatura entre a amostra e a ponta dos termopares.

Os dados de temperatura coletados são exibidos por uma tela de LCD de 2 linhas por 16 colunas acoplada à placa controladora e localizada na parte superior do aparato experimental, mas é possível receber esses dados simultaneamente na tela e em um computador, conectando o microcontrolador Arduino ao computador por meio de interface USB. Conforme definido no código disponível no Apêndice A, o microcontrolador Arduino se comunica com o computador por meio de uma porta serial simulada com velocidade de 115200.

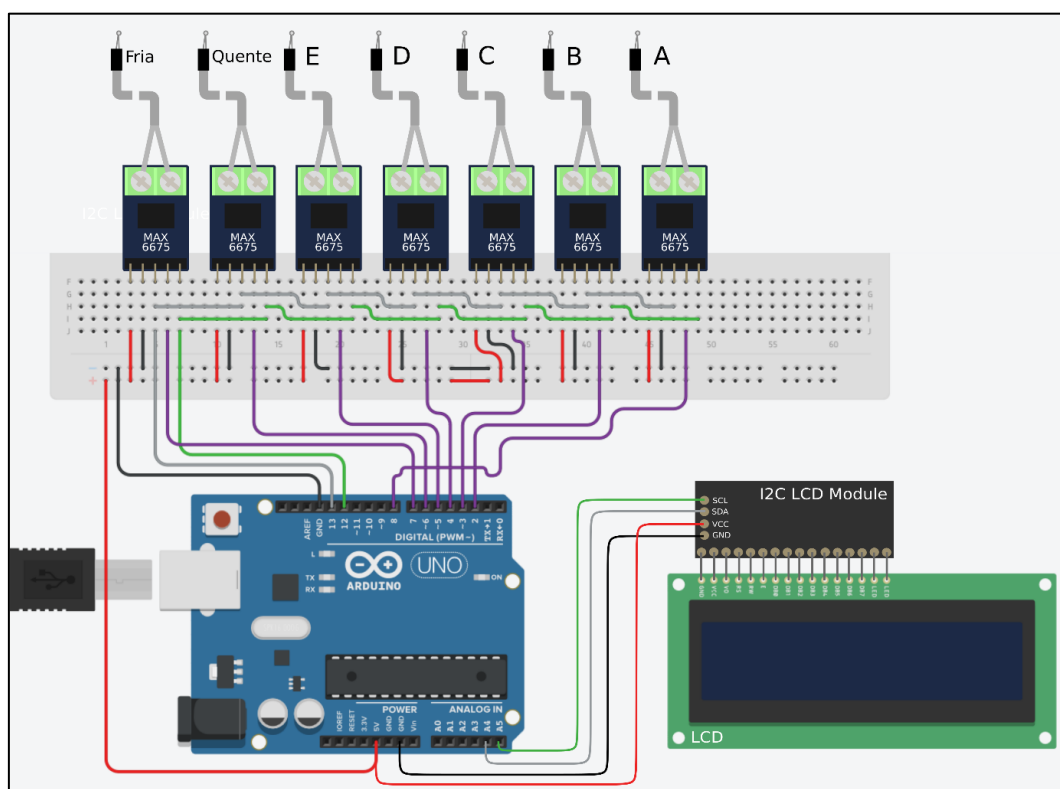
No momento da inicialização do experimento é necessário enviar por meio da conexão serial o número de leituras desejadas e o intervalo em milissegundos entre essas leituras. Após isso o microcontrolador inicia a leitura das temperaturas sequencialmente, porém com um curto período entre a aferição feita por cada um dos termopares. Ao fim, o programa exibe uma lista de todas as medidas realizadas, devidamente classificadas por origem da medição e por ordem de aferição, como mostra a Figura 20, na qual os sensores ainda não haviam sido calibrados.

Figura 20 – Teste dos sensores realizado em temperatura ambiente



A Figura 21 apresenta todos os componentes eletrônicos utilizados no experimento, incluindo as ligações elétricas, porém é importante ressaltar que a *protoboard* foi inclusa apenas para facilitar a representação, já que não faz parte do experimento construído e todo o cabeamento foi soldado diretamente aos componentes.

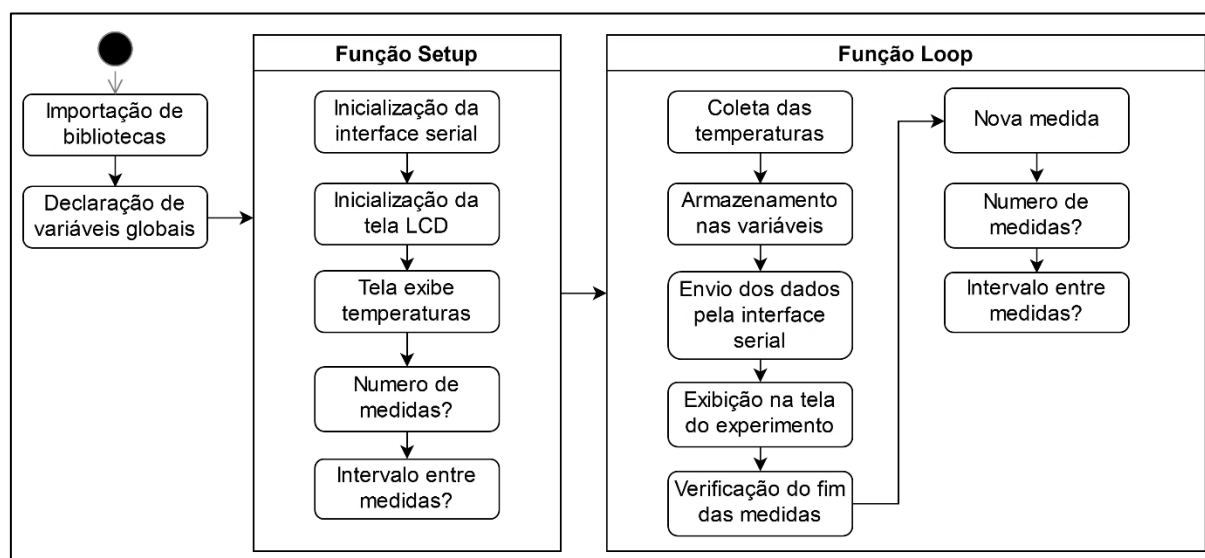
Figura 21 – Diagrama dos componentes eletrônicos do experimento



Durante a execução dos experimentos foi identificado um erro na execução do código do Arduino que em alguns casos provocou alta variação nos dados de leitura. Esse problema ainda não foi solucionado, porém a hipótese plausível para o motivo desse erro é de que *strings* manipuladas na programação provocaram erros no endereçamento de memória.

O diagrama da Figura 22 mostra os principais elementos de código adicionados à programação do Arduino Uno. Destaca-se que a coleta de temperaturas é feita de acordo com a sequência A, B, C, D, E, Fria e Quente, com um intervalo de 40 milissegundos entre elas.

Figura 22 – Funções integradas ao código do Arduino Uno



O Apêndice B compreende a lista de partes do experimento físico de condução de calor. Não foi feita, no entanto, uma estimativa de custo já que muitas partes foram fornecidas pela instituição ou são reutilizadas de outros projetos, porém ficou evidente durante o desenvolvimento do experimento que o conjunto de termopares e de placas MAX6675 foram os itens mais custosos dessa etapa do projeto.

3.2 MODELO COMPUTACIONAL DE CONDUÇÃO TÉRMICA

O modelo computacional foi criado de acordo com alguns requisitos funcionais, como realizar corretamente a simulação do fenômeno físico, permitir que o usuário altere os parâmetros de simulação dinamicamente e possa visualizar as mudanças de

estado da simulação em tempo real. Além disso, as ações do usuário são interpretadas por meio de botões ou outros elementos de interação na interface gráfica.

Portanto, o modelo computacional do experimento de Condução Térmica foi criado no *software NetLogo* com base no método das diferenças finitas, de maneira que os agentes do tipo turtle do NetLogo representem os nós m e n da malha da amostra e os agentes do tipo patch representem as dimensões e o posicionamento da amostra, das fontes de calor e das fontes frias.

As condições de contorno a amostra no modelo são consideradas como isoladas, o que é razoável devido às camadas de isolamento incorporadas ao experimento físico.

A parte central do funcionamento desse modelo consiste no trecho do código⁵ destacado no Quadro 3, no qual cada elemento da amostra requisita a temperatura dos quatro elementos vizinhos e em seguida faz com que a sua própria temperatura seja igual à média dessas temperaturas. Esse é um processo iterativo, que ocorre a cada passagem de tempo da simulação.

Quadro 3 – Código para o cálculo da temperatura

```
ask pedacosdechapa [
  if any? turtles-on neighbors4 and pcolor != red and pcolor != blue
  [
    set temperatura mean [temperatura] of turtles-on neighbors4
    colorir
  ]
]
```

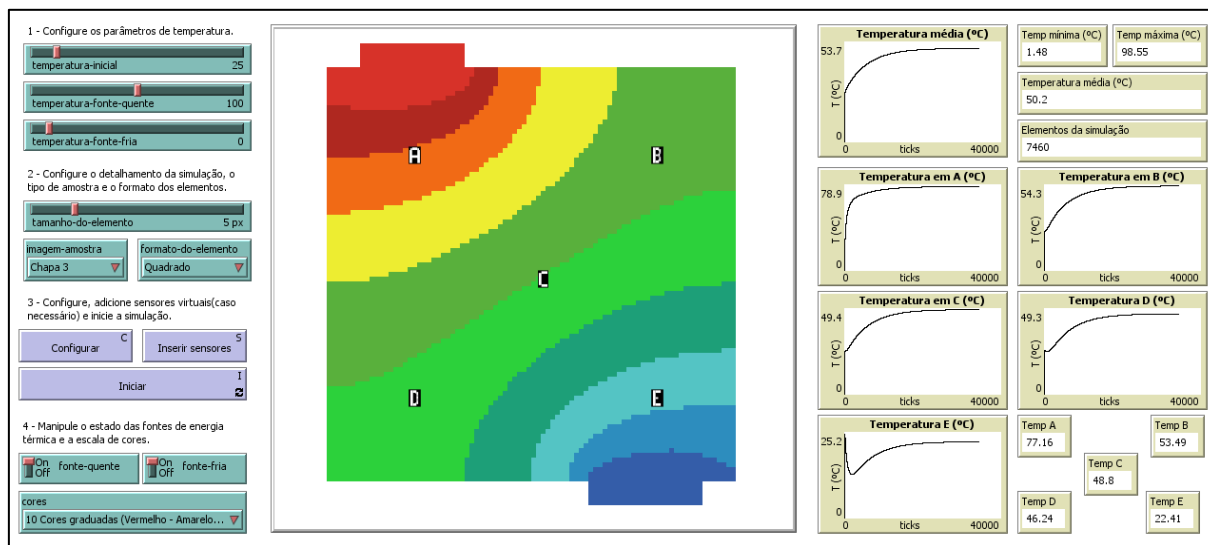
Isso faz com que, por meio de um processo emergente, cada elemento da amostra assuma uma temperatura de maneira condizente com o que ocorre no fenômeno físico, e a partir disso surjam os perfis de temperatura esperados.

O trecho de código do Quadro 3 também mostra uma chamada para o procedimento “colorir”, que, como o nome sugere, altera a cor dos agentes de acordo com a sua temperatura atual segundo uma escala de dez cores predeterminadas, do vermelho ao azul por padrão.

⁵ Nessa seção serão comentados apenas os principais pontos relacionados ao código, no entanto o código completo desse modelo está disponível no Apêndice C.

Essas cores são exibidas na interface do modelo, que foi dividida entre entrada de parâmetros, mundo de simulação e saída de dados, à esquerda, ao centro e à direita, respectivamente. É o que ilustra a Figura 23, uma captura de tela da interface no momento em que a variação de temperatura nos pontos de análise se aproximou o suficiente de zero para que o sistema possa ser considerado em equilíbrio.

Figura 23 – Interface do experimento de condução de calor



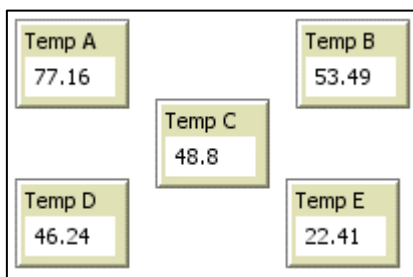
Os parâmetros que podem ser alterados a partir da interface são temperatura da fonte quente, temperatura da fonte fria, temperatura inicial da amostra e o tamanho dos elementos de simulação, o que se assemelha ao refinamento da malha em muitos pacotes de simulação. Também é possível selecionar o estado de ligado e desligado da fonte quente e da fonte fria. Esses parâmetros são então usados no momento da simulação.

A linha formada entre duas cores na amostra é a linha isotérmica, indicando locais de mesma temperatura. Inicialmente esperava-se incluir também as linhas de fluxo de calor, perpendiculares às linhas isotérmicas, porém essa funcionalidade foi descartada.

Assim como no experimento físico, o modelo computacional recebeu sensores virtuais nos pontos A, B, C, D e E, que indicam a temperatura do elemento de amostra abaixo de cada sensor. Com auxílio dos valores de temperatura da fonte fria e da fonte quente, os dados dos sensores virtuais foram usados para as etapas de comparação e validação do modelo computacional.

Os monitores de dados localizados no canto inferior direito da interface apresentam a temperatura de cada um desses sensores virtuais, como detalhado na Figura 24. Também foram inseridos gráficos que mostram a temperatura percebida pelos sensores em relação à passagem de ticks.

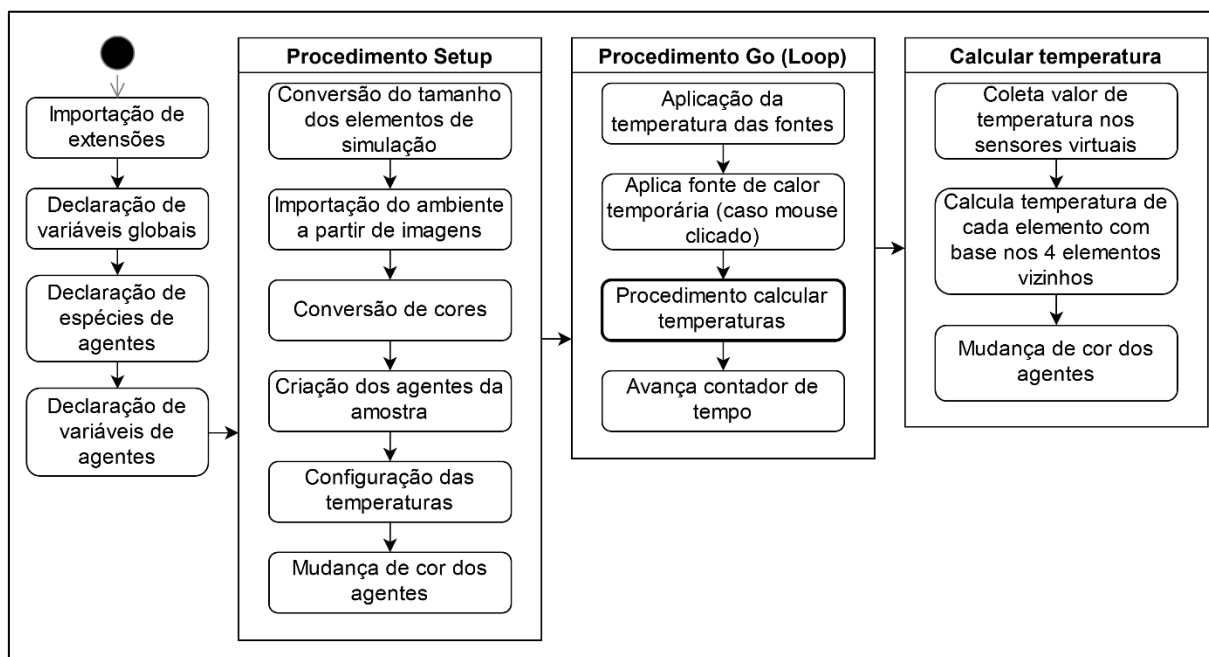
Figura 24 – Monitores de temperatura do modelo



Para efeitos de simplificação, a fonte quente e a fonte fria foram consideradas como seções do ambiente com temperatura uniforme. A temperatura dessas seções pode ser alterada por meio de deslizadores na interface do experimento, influenciando na temperatura dos agentes que representam a chapa.

A Figura 25 apresenta os procedimentos do modelo que são executados durante a simulação. Destaca-se o procedimento do cálculo de temperaturas, parte central do modelo, também responsável pela coleta dos dados pelos sensores virtuais e por invocar o procedimento de mudança de cor dos agentes, que proporciona a visualização do perfil de temperaturas na interface.

Figura 25 – Diagrama de procedimentos do modelo de condução de calor



Uma das funcionalidades implementadas no modelo foi a possibilidade de adicionar à simulação fontes de calor temporárias apenas clicando com o botão esquerdo do mouse. Isso permite visualizar de forma dinâmica o comportamento do perfil de temperatura.

Visto que não é o objetivo deste trabalho analisar a simulação durante sua parte transiente, decidiu-se não usar o tempo real como medida da passagem de tempo do modelo. Ainda assim, seria trabalhoso implementar essa funcionalidade devido a limitações de processamento entre diferentes computadores e à diferença no número de agentes de acordo com os parâmetros de simulação, o que pode fazer com que o modelo seja executado mais rapidamente ou mais lentamente. Também por não ser o objetivo dessa análise avaliar a parte transiente do fenômeno, o coeficiente de condução do material não foi representado.

Por fim, destaca-se que para a validação desse modelo alguns parâmetros como o tamanho dos elementos permaneceram constantes, enquanto outros foram alterados para corresponder aos parâmetros obtidos pelos sensores no experimento físico, como a temperatura ambiente, a temperatura da fonte fria e a temperatura da fonte quente.

3.3 METODOLOGIA ESTATÍSTICA

O modelo foi avaliado segundo a metodologia da Modelagem Bifocal, conforme descrita por Blikstein e Wilensky (2007) e apresentada em forma de diagrama na Figura 7, através de experimentos realizados em laboratório e computacionalmente. A comparação dos resultados foi feita primeiramente por testes t não pareados e posteriormente por testes t pareados.

Como o desvio padrão dos resultados gerados pelo modelo computacional é extremamente pequeno, os testes t não pareados foram feitos para uma amostra, comparando essa amostra ao resultado obtido na simulação computacional.

O nível de significância escolhido para os testes de hipótese foi de 1%, com 0,01 como valor de α e análise bilateral. Para todos os experimentos físicos foram coletadas 31 amostras, usadas nos testes t não pareados. Já os testes t pareados usaram apenas o valor da média dessas amostras na análise.

Nestes testes a hipótese nula H_0 considera convergência dos resultados, enquanto que a hipótese alternativa H_1 considera divergência dos resultados. Ou seja, para os testes foi considerado que $H_0: \mu_1 = \mu_2$ e que $H_1: \mu_1 \neq \mu_2$.

4 RESULTADOS E DISCUSSÃO

Inicialmente foi feita uma calibração dos sensores do experimento físico em relação a termômetros analógicos de vidro do Laboratório de Física da Universidade do Vale do Itajaí. O valor de temperatura ambiente médio observado entre os termômetros foi de 24,57 °C.

Portanto, foram feitas 31 leituras dos sensores do experimento em temperatura ambiente, com um intervalo de 1 segundo entre cada, a fim de obter a média do valor de cada sensor e assim calcular a diferença entre a leitura dos termômetros e a dos sensores. Os valores obtidos para essa diferença são disponibilizados na Tabela 1.

Tabela 1 – Medida para calibração dos sensores

Dados	A (°C)	B (°C)	C (°C)	D (°C)	E (°C)	Fria (°C)	Quente (°C)
Média dos termopares	26,02	25,86	26,09	25,85	26,60	25,64	26,13
Desvio Padrão	0,1573	0,2125	0,2194	0,1789	0,1680	0,2313	0,1923
Diferença em relação a ambiente	1,44	1,29	1,52	1,28	2,03	1,07	1,56

É evidente na Tabela 1 que todos os sensores de temperatura do experimento superestimam a temperatura aferida em relação aos termômetros do laboratório. Portanto, o valor de diferença para cada sensor foi subtraído dos dados coletados no restante das medições.

Com a calibração dos sensores foi possível realizar as medidas com diferentes temperaturas para fonte fria e fonte quente. Uma das medidas realizadas foi descartada por apresentar um coeficiente de variação de 0,13 para os dados da fonte fria, um valor considerado alto em relação às outras medidas e que indica que houve problemas como derretimento em excesso ou deslocamento do gelo na fonte fria.

Outras 11 medidas foram descartadas pois um defeito na fonte de alimentação utilizada inicialmente provocou alta variação nos dados coletados na fonte quente, algo que certamente prejudicaria a análise. A fonte de alimentação danificada foi substituída e novos dados foram coletados.

Para a primeira medida considerada válida, mostrada na Tabela 2 juntamente com as outras medidas, a tensão aplicada ao resistor foi de 1,3 V e a corrente aplicada foi de 1,3 A, fornecendo ao resistor 1,69 W de potência, aquecendo a amostra de tal

forma que durante a leitura a média de temperatura da fonte quente atingiu 53,44 °C. Já o gelo da fonte fria fez com que a temperatura mensurada nesse ponto atingisse 1,77 °C.

Tabela 2 – Dados de temperatura coletados no experimento físico

Medida	Dados	A (°C)	B (°C)	C (°C)	D (°C)	E (°C)	Fria (°C)	Quente (°C)
Medida 1	Média	40,92	31,86	29,25	25,97	22,37	1,77	53,44
	Desvio padrão	0,3971	0,2714	0,1700	0,2582	0,2790	0,2376	0,3354
Medida 2	Média	45,22	35,17	32,71	28,68	23,76	4,10	59,16
	Desvio padrão	0,3947	0,3102	0,2983	0,3056	0,2336	0,3439	0,3913
Medida 3	Média	49,38	40,50	32,56	27,52	23,03	1,94	66,89
	Desvio padrão	0,2677	0,2821	0,2345	0,3056	0,2579	0,2540	0,2792
Medida 4	Média	53,04	41,01	33,00	32,78	23,18	1,81	75,25
	Desvio padrão	0,2734	0,2938	0,2953	0,3692	0,1720	0,2404	0,3143
Medida 5	Média	48,07	33,93	31,05	27,35	23,72	1,97	81,64
	Desvio padrão	0,2656	0,2304	0,2579	0,2572	0,2187	0,3010	0,2845
Medida 6	Média	60,26	39,36	35,72	29,35	24,01	1,77	99,50
	Desvio padrão	0,2535	0,3396	0,2090	0,2651	0,1981	0,2996	0,4459

A medida 2 foi realizada com 1,4 V e 1,4 A aplicados ao resistor, fornecendo uma potência de 1,96 W, suficiente para aquecer a fonte quente à 59,16 °C, enquanto que a temperatura observada na fonte fria foi de 4,10 °C.

Já na medida 3, com 1,7 V e 1,7 A, ou seja, 2,89 W colocados no resistor, os dados mensurados para fonte quente e para fonte fria são respectivamente 66,89 °C e 1,94 °C. A medida 4 foi realizada com uma maior potência aplicada ao resistor, de 3,24 W, com 1,8 V e 1,8 A fornecidos pela fonte. Nesse caso, a medida de temperatura da fonte quente foi de 75,25 °C e a medida registrada pela fonte fria foi de 1,81 °C.

Na sequência, a medida 5 usou como valores de tensão e corrente 2 V e 2 A, resultando em uma potência de 4 W. Esse aumento na potência levou a fonte quente a uma temperatura de 81,64 °C e a fonte fria a 1,97 °C. No entanto, observou-se uma inconsistência em relação à medida anterior, que apresenta valores de temperatura maiores mesmo a fonte quente estando a uma menor temperatura.

Por fim, a sexta e última medida válida registrada foi feita com 2,3 V e 2,3 A aplicados ao resistor, com uma potência de 5,29 W. Esses parâmetros fizeram com que a temperatura da fonte quente atingisse 99,50 °C, enquanto que a temperatura registrada na fonte fria foi de 1,77 °C.

Após a coleta dos dados do modelo físico foi possível aplicar os dados de temperatura obtidos ao modelo computacional. Assim como no experimento físico, foram feitas 31 medidas, porém neste caso reiniciando a simulação e esperando que atingisse a estabilidade a cada medida.

Os parâmetros fixos configurados para a simulação foram o tamanho dos elementos, selecionado como 8 pixels, e a temperatura inicial, mensurada como 24,57 °C no experimento físico. Já os parâmetros de temperatura da fonte quente e temperatura da fonte fria foram alterados para corresponder a cada uma das medidas realizadas no experimento físico.

Algo que se observou foi que, diferentemente do que ocorreu no experimento físico, o desvio padrão calculado a partir dos dados dos sensores virtuais foi extremamente baixo, sendo que o maior valor encontrado foi de 0,0007, mostrando que o modelo computacional tem precisão em relação à saída dos dados.

Portanto, decidiu-se apresentar todos os dados obtidos na Tabela 3, na qual cada medida foi identificada numericamente e o desvio padrão foi incluído na linha abaixo. Os valores da fonte quente e da fonte fria para cada medida são os mesmos obtidos no experimento físico, sendo também apresentados na tabela.

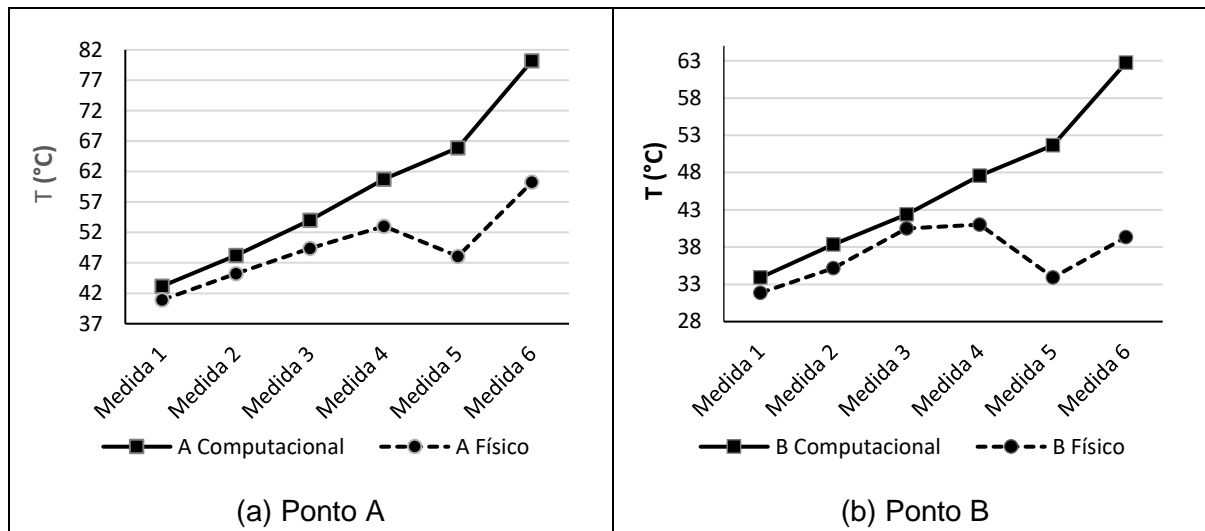
Tabela 3 – Dados de temperatura coletados na simulação computacional

Medida	Dados	A (°C)	B (°C)	C (°C)	D (°C)	E (°C)	Fria (°C)	Quente (°C)
Medida 1	Média	43,19	33,91	29,35	26,31	15,83	1,77	53,44
	Desvio padrão	0,0004	0,0007	0,0006	0,0007	0,0004	-	-
Medida 2	Média	48,25	38,36	33,50	30,26	19,09	4,10	59,16
	Desvio padrão	0,0003	0,0005	0,0005	0,0006	0,0003	-	-
Medida 3	Média	54,04	42,40	36,66	32,85	19,65	1,94	66,89
	Desvio padrão	0,0003	0,0006	0,0005	0,0006	0,0004	-	-
Medida 4	Média	60,74	47,59	41,10	36,79	21,85	1,81	75,25
	Desvio padrão	0,0004	0,0007	0,0006	0,0007	0,0004	-	-
Medida 5	Média	65,91	51,65	44,61	39,94	23,73	1,97	81,64
	Desvio padrão	0,0005	0,0008	0,0008	0,0009	0,0006	-	-
Medida 6	Média	80,23	62,76	54,12	48,40	28,49	1,77	99,50
	Desvio padrão	0,0003	0,0006	0,0006	0,0007	0,0004	-	-

A partir desses dados é possível observar que os valores das Medida 1 e 2 foram os mais próximos dos valores mensurados no experimento físico. No entanto as demais medidas apresentaram desvios consideráveis entre o experimento físico e o modelo computacional, o que é perceptível nos gráficos da Figura 26, nos quais os

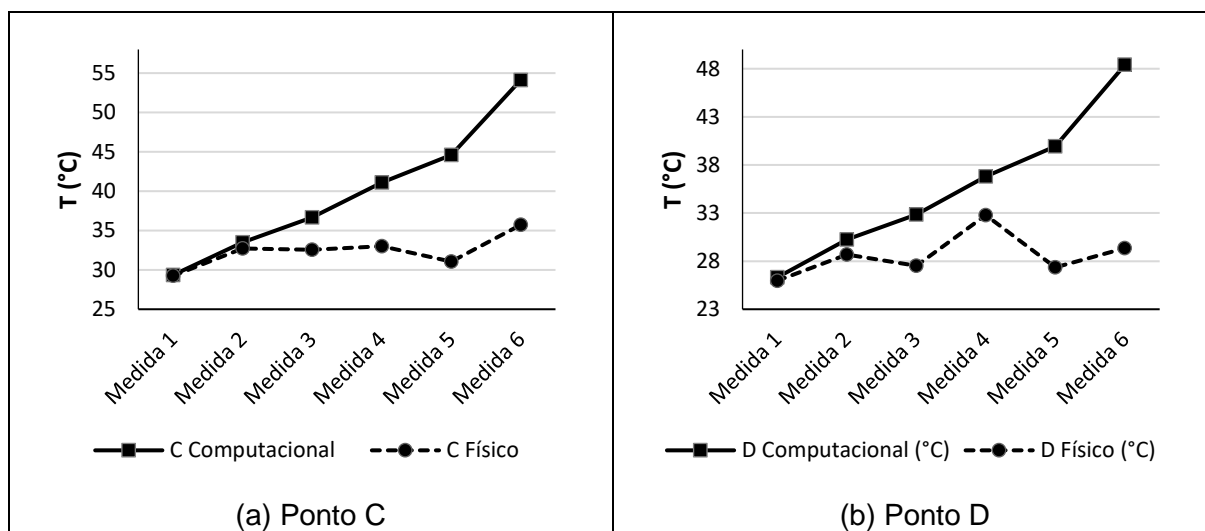
valores físicos e computacionais mensurados nos pontos A e B começam próximos e vão divergindo com o passar das medidas.

Figura 26 – Comparação dos dados físicos e computacionais nos pontos A e B



Da mesma forma os dados dos pontos C e D, representados nos gráficos da Figura 27, iniciaram muito próximos, se afastando ao longo das medidas. Nota-se uma queda no valor de temperatura na medida 5 em relação à medida 4 do experimento físico, o que pode indicar uma inconsistência na aplicação das fontes de calor.

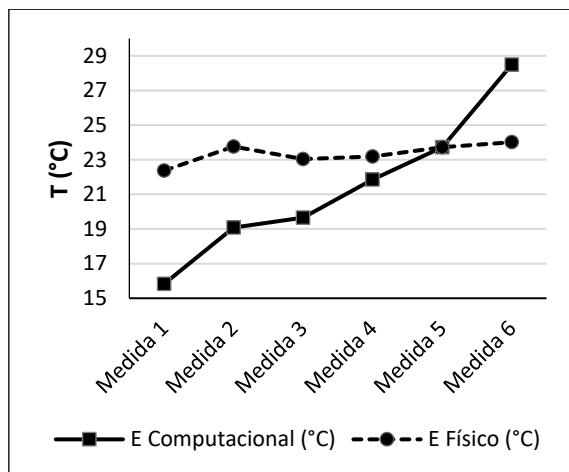
Figura 27 – Comparação dos dados físicos e computacionais nos pontos C e D



No ponto E, representado na Figura 28, algo diferente ocorreu. Logo na primeira medida os valores já são distantes, sendo que enquanto o valor computacional cresceu conforme esperado pelo aquecimento progressivo da fonte quente ao longo das medidas, os valores obtidos do experimento físico permaneceram

quase que constantes. Pode-se supor que nesse caso o termopar não estava devidamente encostado na amostra, causando a diferença observada.

Figura 28 – Comparação dos dados físicos e computacionais no ponto E



Portanto, foram criadas algumas hipóteses para o motivo dessas diferenças. A primeira delas é a de que erros no momento do experimento físico podem ter provocado variação nos dados. O método de resfriamento da fonte fria utilizado pode causar essa variação pelo derretimento do gelo, assim como a infiltração de água pode alterar a temperatura da amostra. No entanto houve uma tentativa de minimizar esses fatores durante o experimento.

Também é importante levar em consideração que a calibração dos sensores foi feita apenas com base na temperatura ambiente e é possível que em outras faixas de temperatura exista uma diferença relevante entre o valor mensurado e o valor real. Logo, a fim de anular esse efeito deve ser realizado um estudo de calibração desses sensores em uma ampla faixa de temperaturas.

Outro ponto a se considerar é a perda de calor para o ambiente, mesmo com todo o isolamento colocado em torno da amostra. Como o fluxo de calor é uma função da diferença de temperatura, a perda ou ganho de calor da amostra pode ocorrer com maior intensidade nas regiões onde a temperatura é mais alta ou mais baixa em relação à temperatura ambiente. Uma solução possível, mas não viável no contexto do trabalho, seria realizar os ensaios em uma câmara a vácuo, anulando fatores como convecção do ambiente, mas ainda sofrendo os efeitos de radiação.

A solução adotada para minimizar essa divergência entre os dados foi substituir tanto a fonte quente como a fonte fria por pastilhas termoelétricas de Peltier, anulando

de uma só vez problemas como o derretimento do gelo, a infiltração de água na amostra e possíveis erros de leitura devido ao contato direto do gelo com o termopar da fonte fria. Para aprimorar ainda mais a confiabilidade do experimento o tempo de estabilização da amostra foi aumentado para dez minutos.

As pastilhas de Peltier usadas são do modelo TEC1-12715, com 40 mm de largura, 40 mm de altura, 3 mm de espessura e capacidade de produzir uma diferença de temperatura máxima entre as faces de 70 °C, aceitando valores máximos de corrente e de tensão de 15 A e 15,4 V. (HB, 2018).

Os dados obtidos nesse experimento reformulado são apresentados na Tabela 4. Porém, por limitações da faixa de temperatura de operação das pastilhas de Peltier foram realizadas apenas 4 medidas com a fonte quente entre 40,36 °C e 70,23 °C. Medidas preliminares foram feitas para garantir que todos os termopares estavam em contato com a amostra e registrando sua temperatura adequadamente.

Tabela 4 – Dados de temperatura coletados no experimento físico reformulado

Medida	Dados	A (°C)	B (°C)	C (°C)	D (°C)	E (°C)	Fria (°C)	Quente (°C)
Medida 1	Média	30,64	24,42	22,77	18,90	12,76	5,27	40,36
	Desvio padrão	0,2842	0,2245	0,2508	0,2160	0,2365	0,3739	0,2178
Medida 2	Média	38,53	30,32	27,15	21,76	13,75	3,98	50,49
	Desvio padrão	0,2269	0,2221	0,2178	0,2940	0,2719	0,3618	0,2616
Medida 3	Média	44,85	34,26	30,83	24,16	15,29	2,07	60,89
	Desvio padrão	0,2590	0,2275	0,2206	0,2579	0,2254	0,2649	0,2275
Medida 4	Média	52,95	40,76	36,32	28,89	20,43	5,08	70,23
	Desvio padrão	0,3012	0,3254	0,2780	0,2178	0,3145	0,3340	0,2746

Novas simulações foram executadas com base nos parâmetros de temperatura coletados no experimento físico reformulado e foram obtidos novos dados do modelo computacional, apresentados na Tabela 5.

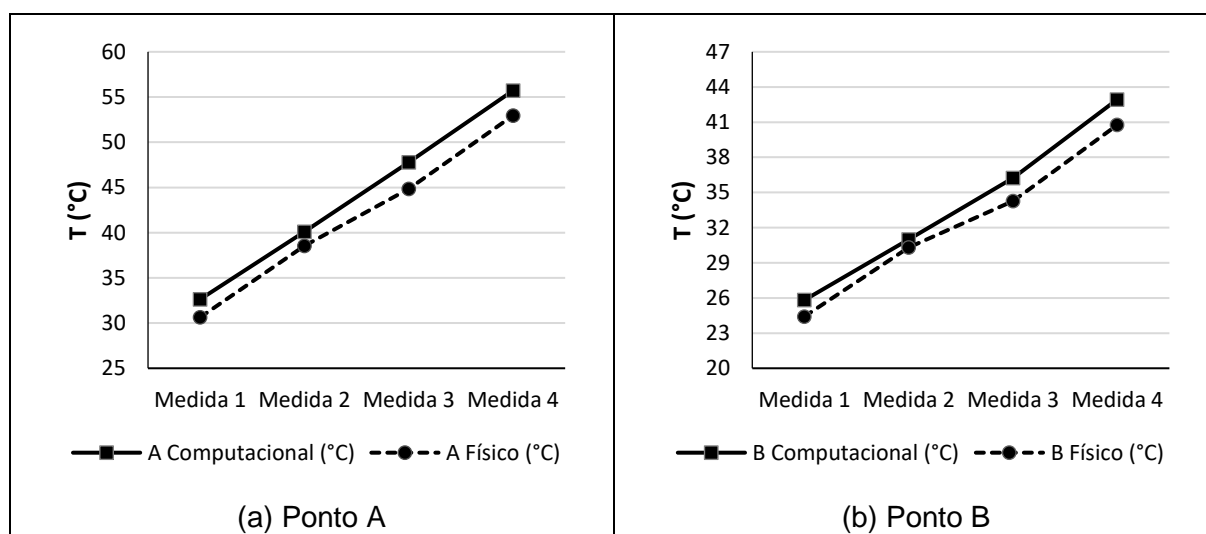
Tabela 5 – Dados de temperatura coletados na simulação computacional

Medida	Dados	A (°C)	B (°C)	C (°C)	D (°C)	E (°C)	Fria (°C)	Quente (°C)
Medida 1	Média	32,63	25,82	22,25	19,39	12,46	5,27	40,36
	Desvio padrão	0,0002	0,0003	0,0003	0,0003	0,0002	-	-
Medida 2	Média	40,10	30,98	26,26	22,46	13,39	3,98	50,49
	Desvio padrão	0,0002	0,0003	0,0003	0,0003	0,0002	-	-
Medida 3	Média	47,77	36,24	30,28	25,46	13,98	2,07	60,89
	Desvio padrão	0,0002	0,0004	0,0003	0,0004	0,0002	-	-
Medida 4	Média	55,71	42,94	36,33	31,00	18,28	5,08	70,23
	Desvio padrão	0,0002	0,0003	0,0003	0,0003	0,0002	-	-

Ao longo das simulações ficou claro que o NetLogo executou o modelo de maneira mais lenta do que ocorre no fenômeno real, sendo que com 2916 agentes cada *tick* levou em média 0,05 segundos para ser processado, com um total de 6,04 minutos para a conclusão de cada simulação.

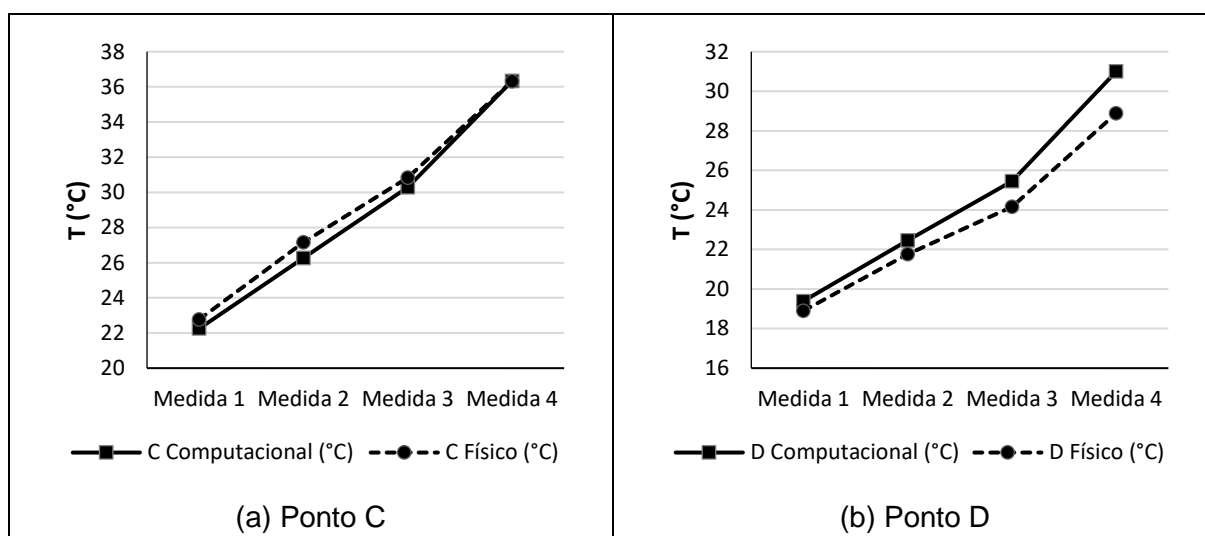
É perceptível que os dados do modelo em conjunto com o experimento reformulado foram muito mais condizentes que os obtidos anteriormente. A Figura 29 apresenta esses dados em forma de gráfico para os pontos A e B, nos quais é visível que os dados do experimento físico foram muito mais coerentes com o que se espera que aconteça no fenômeno de condução de calor. Quando comparados ponto a ponto, os dados experimentais e computacionais obtidos ficaram relativamente próximos, com uma diferença que pode ser explicada pela perda de calor.

Figura 29 – Comparação dos dados em A e B com experimento reformulado



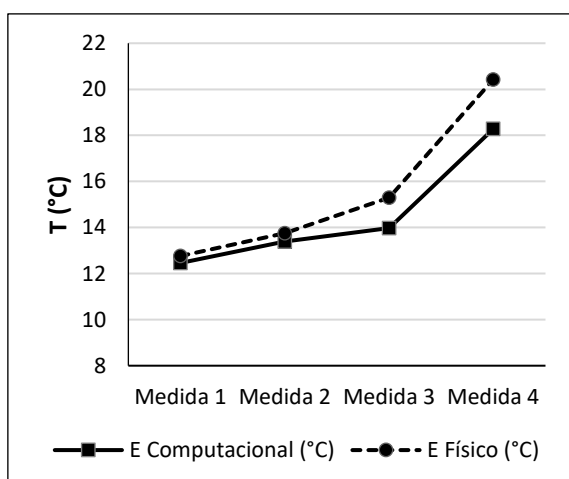
Da mesma forma, as temperaturas obtidas nos pontos C e D se mostraram bastante próximas entre modelo e experimento. Entre todos os pontos, o ponto C foi o que apresentou os resultados mais próximos entre o modelo computacional e o experimento físico, possivelmente por sua localização proporcionar uma menor perda de calor para o ambiente. Já o ponto D apresentou uma leve divergência ao passar das medidas.

Figura 30 – Comparação dos dados em C e D com experimento reformulado



O ponto E, mais próximo da fonte fria, apresentou um comportamento distinto dos anteriores, com a temperatura do experimento físico acima da observada nas simulações. Isso pode estar relacionado com o fato de a temperatura nessa região estar abaixo da temperatura ambiente em todas as medidas, logo é plausível afirmar que houve um ganho de calor do ambiente nessa região.

Figura 31 – Comparação dos dados em E com experimento reformulado



Como precaução, o modelo foi revisado em busca de erros ou divergência em relação à teoria, porém ficou evidente que o código utilizado no modelo segue os mesmos métodos disponíveis na literatura. É notável, portanto, a necessidade de um fator de correção para mitigar essa diferença entre o experimento e o modelo.

Esse fator foi desenvolvido de maneira a representar a condução de calor que pode ocorrer entre a placa e o restante da estrutura do experimento, mesmo que essa estrutura seja feita de material isolante. Em código, conforme destacado no Quadro 4, a cada *tick* da simulação o valor do fator de correção, multiplicado pela centésima parte do valor da temperatura do agente, é subtraído da temperatura do elemento, caso essa temperatura seja maior que a temperatura ambiente, ou somado, caso a temperatura seja menor que a temperatura ambiente.

Quadro 4 – Cálculo de temperatura reformulado

```
ask pedacosdechapa [
  if any? turtles-on neighbors4 and pcolor != red and pcolor != blue
  [
    set temperatura mean [temperatura] of turtles-on neighbors4
    ;fator de correção
    ifelse temperatura > temperatura-media [
      set temperatura (temperatura - fator-de-correcao * temperatura / 100)
    ] [
      set temperatura (temperatura + fator-de-correcao * temperatura / 100)
    ]
    colorir
  ]
]
```

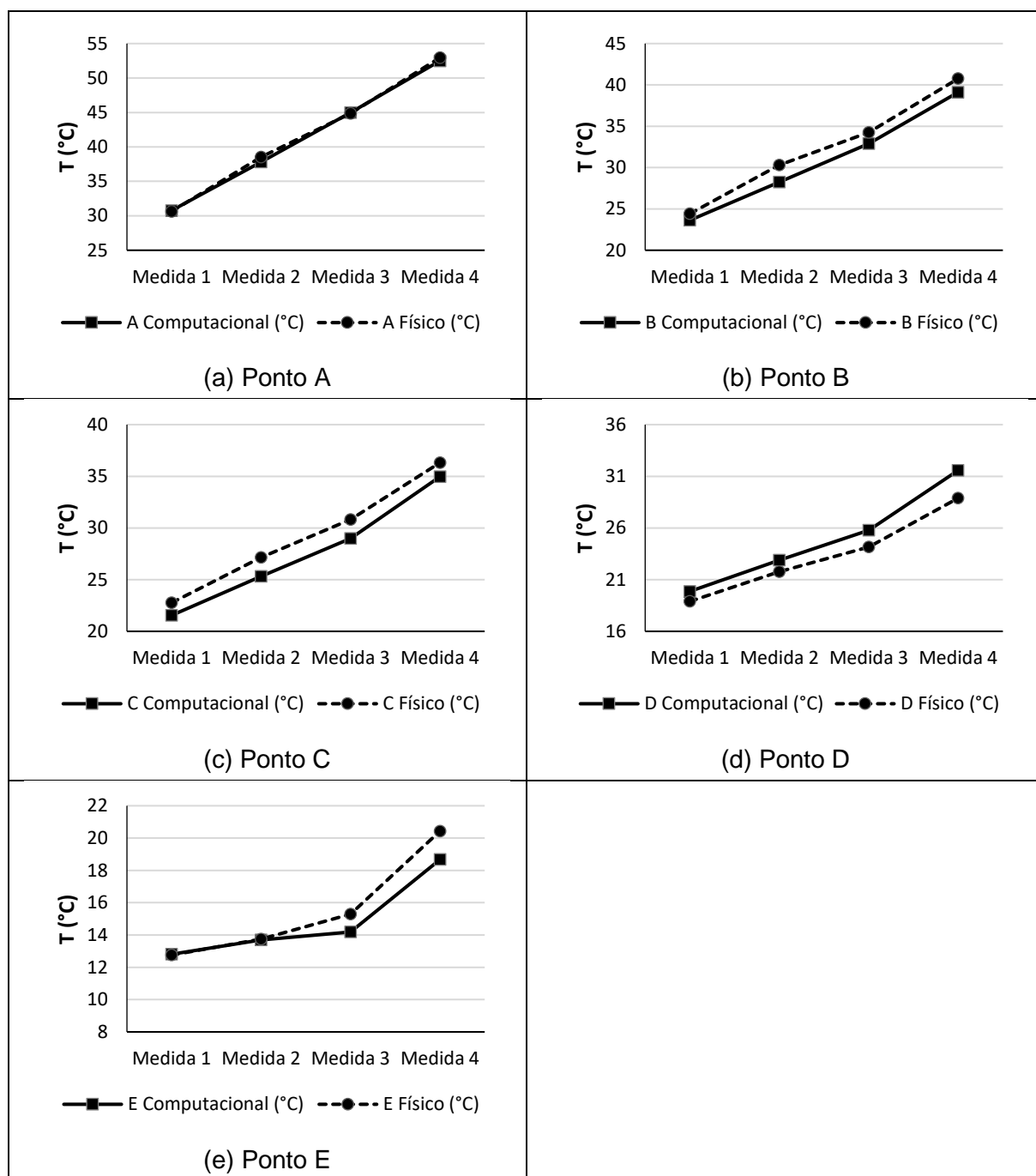
O valor do fator de correção que melhor correspondeu aos testes foi de 0,01, com o qual foram executadas novas simulações do modelo. Os dados obtidos nessas simulações são disponibilizados na Tabela 6 e tornam perceptível que o fator de correção trouxe maior proximidade entre os dados do modelo e do experimento físico, porém é importante ressaltar que esse fator não elimina possíveis erros experimentais ou a necessidade de melhorias no experimento.

No entanto, como pode ser notado nos gráficos da Figura 32, enquanto os dados dos pontos A e E se aproximaram das medidas obtidas no experimento físico após a inclusão do fator de correção, os dados dos pontos C, D e E acabaram se afastando da medida física.

Tabela 6 – Dados da simulação com o fator de correção

Medida	A (°C)	B (°C)	C (°C)	D (°C)	E (°C)	Fria (°C)	Quente (°C)
Medida 1	30,77	23,63	21,55	19,87	12,81	5,27	40,36
Medida 2	37,80	28,24	25,31	22,90	13,69	3,98	50,49
Medida 3	44,98	32,89	28,99	25,79	14,18	2,07	60,89
Medida 4	52,49	39,11	34,97	31,58	18,67	5,08	70,23

Figura 32 – Dados experimentais e computacionais após fator de correção



Ainda assim, a partir desses dados foram realizados testes estatísticos em comparação aos dados obtidos no experimento físico levando em conta o nível de significância de 1%. Os testes foram feitos para cada ponto e a cada medida, num total de 20 testes t não pareados, apresentados em formato de tabela a seguir.

4.1 TESTES DE HIPÓTESE INDEPENDENTES

Para a medida 1, representada na Tabela 7, os dados dos pontos A e E se mostraram estatisticamente semelhantes. No entanto, os pontos B, C e D não apresentaram essa semelhança, sendo que para esses pontos a hipótese alternativa foi aceita. Ainda assim, essa diferença das médias é pequena, sendo que futuros refinamentos podem auxiliar nos resultados do modelo e dos experimentos.

Tabela 7 – Testes de hipótese da medida 1

Sensor	A	B	C	D	E
Média computacional	30,77	23,63	21,55	19,87	12,81
Média experimental	30,64	24,42	22,77	18,90	12,76
Desvio padrão experimental	0,2842	0,2245	0,2508	0,2160	0,2365
N	31	31	31	31	31
Graus de liberdade	30	30	30	30	30
Valor t	-2,6829	19,5301	27,2220	-25,0290	-1,1708
Valor P	0,0118	0,0000	0,0000	0,0000	0,2509
t crítico bicaudal	2,7500	2,7500	2,7500	2,7500	2,7500
Hipótese aceita	H0	H1	H1	H1	H0

Da mesma forma, na segunda medida a maior parte dos dados apresentou diferença de médias, sendo a hipótese alternativa aceita nos pontos A, B, C e D. Já o ponto E apresentou igualdade entre as médias. Algo notável é que o desvio padrão experimental é pequeno, logo diferenças de aproximadamente 1 °C, como as observadas entre os experimentos, serão consideradas estatisticamente diferentes.

Tabela 8 – Testes de hipótese da medida 2

Sensor	A	B	C	D	E
Média computacional	37,80	28,24	25,31	22,90	13,69
Média experimental	38,53	30,32	27,15	21,76	13,75
Desvio padrão experimental	0,2269	0,2221	0,2178	0,2940	0,2719
N	31	31	31	31	31
Graus de liberdade	30	30	30	30	30
Valor t	17,9411	52,1790	47,0855	-21,6492	-0,0236
Valor P	0,0000	0,0000	0,0000	0,0000	0,9813
t crítico bicaudal	2,7500	2,7500	2,7500	2,7500	2,7500
Hipótese aceita	H1	H1	H1	H1	H0

Já na medida 3 todos os pontos apresentaram diferença entre os dados, sendo a hipótese alternativa aceita. No entanto, os dados do ponto A ficaram extremamente próximos de ser considerados estatisticamente semelhantes, visto que o valor de teste -2,8890 é próximo, mas ultrapassa o valor crítico -2,75.

Tabela 9 – Testes de hipótese da medida 3

Sensor	A	B	C	D	E
Média computacional	44,98	32,89	28,99	25,79	14,18
Média experimental	44,85	34,26	30,83	24,16	15,29
Desvio padrão experimental	0,2590	0,2275	0,2206	0,2579	0,2254
N	31	31	31	31	31
Graus de liberdade	30	30	30	30	30
Valor t	-2,8890	33,4563	46,4313	-35,1816	27,3981
Valor P	0,0071	0,0000	0,0000	0,0000	0,0000
t crítico bicaudal	2,7500	2,7500	2,7500	2,7500	2,0423
Hipótese aceita	H1	H1	H1	H1	H1

Por fim, na medida 4 todas os testes apontaram para a validade da hipótese alternativa de diferença de médias, porém dessa vez com uma diferença mais expressiva, com valores de teste muito distantes do valor crítico. Novamente, o ponto com dados mais próximos foi o ponto A, com valor de teste 8,5115, porém superior ao valor crítico 2,75.

Tabela 10 – Testes de hipótese da medida 4

Sensor	A	B	C	D	E
Média computacional	52,49	39,11	34,97	31,58	18,67
Média experimental	52,95	40,76	36,32	28,89	20,43
Desvio padrão experimental	0,3012	0,3254	0,2780	0,2178	0,3145
N	31	31	31	31	31
Graus de liberdade	30	30	30	30	30
Valor t	8,5115	28,1757	27,0662	-68,6569	31,2326
Valor P	0,0000	0,0000	0,0000	0,0000	0,0000
t crítico bicaudal	2,7500	2,7500	2,7500	2,7500	2,7500
Hipótese aceita	H1	H1	H1	H1	H1

Portanto, rejeita-se a hipótese H_0 de convergência entre o modelo computacional e o experimento físico nos testes de hipótese não pareados, feitos ponto a ponto e medida a medida, ou seja, de acordo com a maior parte das medidas o modelo apresenta divergência em relação aos dados coletados experimentalmente nesses testes.

4.2 TESTES DE HIPÓTESE PAREADOS

Também foram feitos testes de hipótese pareados para cada ponto de coleta, associando os valores obtidos nas medidas 1, 2, 3 e 4 do experimento físico e computacional. Os dados utilizados para o teste t pareado no ponto A são apresentados na Tabela 11.

Tabela 11 – Dados do ponto A emparelhados

Sensor	A Computacional (°C)	A Físico (°C)
Medida 1	30,77	30,64
Medida 2	37,8	38,53
Medida 3	44,98	44,85
Medida 4	52,49	52,95

O resultado do teste t pareado para o ponto A, conforme a Tabela 12, indica que a hipótese aceita é H_0 , de igualdade. Portanto, para este ponto pode-se considerar que há validade estatística entre o modelo e o experimento, com forte correlação entre os dados.

Tabela 12 – Teste t pareado para os dados do ponto A

Ponto A	Computacional	Físico
Média (°C)	41,51	41,74
Variância	87,2377	89,6168
Observações	4	4
Correlação de Pearson	0,9990	
Graus de liberdade	3	
Valor t	-1,0743	
Valor P	0,3614	
t crítico bicaudal	5,8409	
Hipótese aceita	H0	

Já os dados utilizados para o teste t pareado no ponto B são apresentados na Tabela 13.

Tabela 13 – Dados do ponto B emparelhados

Sensor	B Computacional (°C)	B Físico (°C)
Medida 1	23,63	24,42
Medida 2	28,24	30,32
Medida 3	32,89	34,26
Medida 4	39,11	40,76

Assim como no caso do ponto A, os resultados do teste t pareado para os dados do ponto B indicam que há validade estatística entre o experimento e o modelo, com a hipótese H0 aceita e com forte correlação entre os dados.

Tabela 14 – Teste t pareado para os dados do ponto B

Ponto B	Computacional	Físico
Média (°C)	30,97	32,44
Variância	43,7582	47,1165
Observações	4	4
Correlação de Pearson	0,9975	
Graus de liberdade	3	
Valor t	-5,4472	
Valor P	0,0122	
t crítico bicaudal	5,8409	
Hipótese aceita	H0	

A Tabela 15 contém os dados do ponto C emparelhados. Conforme o teste estatístico, esses dados apresentam forte correlação, de 0,9984.

Tabela 15 – Dados do ponto C emparelhados

Sensor	C Computacional (°C)	C Físico (°C)
Medida 1	21,55	22,77
Medida 2	25,31	27,15
Medida 3	28,99	30,83
Medida 4	34,97	36,32

Ainda assim, o teste t pareado mostra que a hipótese aceita para o ponto C é a de diferença, H1, com valor t de -9,6215 menor do que o valor crítico bicaudal 5,8409.

Tabela 16 – Teste t pareado para os dados do ponto C

Ponto C	Computacional	Físico
Média (°C)	27,71	29,27
Variância	32,6838	32,9602
Observações	4	4
Correlação de Pearson	0,9984	
Graus de liberdade	3	
Valor t	-9,6215	
Valor P	0,0024	
t crítico bicaudal	5,8409	
Hipótese aceita	H1	

Os dados do ponto D são disponibilizados na Tabela 17. Como nos casos anteriores também há uma forte correlação entre os dados, de 0,9995.

Tabela 17 – Dados do ponto D emparelhados

Sensor	D Computacional (°C)	D Físico (°C)
Medida 1	19,87	18,9
Medida 2	22,9	21,76
Medida 3	25,79	24,16
Medida 4	31,58	28,89

Neste caso, o resultado do teste t pareado aponta para uma igualdade entre o modelo e o experimento no ponto de coleta D, com a hipótese H0 aceita, conforme a Tabela 18.

Tabela 18 – Teste t pareado para os dados do ponto D

Ponto D	Computacional	Físico
Média (°C)	25,04	23,43
Variância	24,8808	17,8848
Observações	4	4
Correlação de Pearson	0,9995	
Graus de liberdade	3	
Valor t	4,1537	
Valor P	0,0254	
t crítico bicaudal	5,8409	
Hipótese aceita	H0	

Por fim, os dados emparelhados do ponto E são apresentados na Tabela 19 e apresentam uma forte correlação, de 0,9931.

Tabela 19 – Dados do ponto E emparelhados

Sensor	E Computacional (°C)	E Físico (°C)
Medida 1	12,81	12,76
Medida 2	13,69	13,75
Medida 3	14,18	15,29
Medida 4	18,67	20,43

A hipótese aceita no teste t pareado feito com esses dados é a de igualdade, H0, corroborando com a validade do modelo computacional em relação ao experimento físico.

Tabela 20 – Teste t pareado para os dados do ponto E

Ponto E	Computacional	Físico
Média (°C)	14,84	15,56
Variância	6,8493	11,6353
Observações	4	4
Correlação de Pearson	0,9931	
Graus de liberdade	3	
Valor t	-1,6583	
Valor P	0,1958	
t crítico bicaudal	5,8409	
Hipótese aceita	H0	

5 CONCLUSÃO

É possível afirmar que o modelo computacional criado apresentou validade estatística em relação ao experimento físico em quatro dos cinco pontos coleta de dados quando avaliado por testes *t* pareados. Ainda assim, necessita ser revisado para que se tenha confiança nos dados obtidos em relação ao ponto C, onde houve divergência.

Observa-se que, no caso dos testes *t* independentes, mesmo as medidas que não obtiveram convergência estatística ficaram próximas, portanto deve-se investigar as falhas experimentais e de modelo a fim de minimizar os desvios observados.

Alguns fatores que podem ter influenciado negativamente nos resultados são o método de fixação dos termopares, o material de isolamento e o material da própria amostra, que tem baixo coeficiente de condução térmica em relação a materiais como cobre. Portanto, a substituição do material da amostra por um melhor condutor térmico pode levar a resultados mais consistentes.

A falha que ocorre em algumas das inicializações dos sensores ainda não foi corrigida. Uma investigação rápida indica que está relacionada a um erro de endereçamento de memória produzido pela manipulação de *strings* no programa. A simples reinicialização do experimento resolve temporariamente o problema, porém para que este fator seja eliminado e o experimento tenha maior confiabilidade é necessário corrigir essa falha.

Apesar disso, é notável pela teoria e pelo desenvolvimento do trabalho que o método de Modelagem Bifocal, assim como outros métodos, capacita o pesquisador a pensar criticamente sobre seu experimento por sua característica iterativa e de busca por defeitos do experimento ou modelo, exigindo, como no caso deste estudo, reformulações tanto no método experimental quanto no modelo em si em caso de divergência entre os resultados.

Ainda assim é possível argumentar que essa característica iterativa é inerente ao método científico, porém a Modelagem Bifocal se orienta em um campo muito mais educacional e de incentivo ao uso do método científico do que apenas a sua utilização cotidiana. É importante que, principalmente no momento presente, iniciativas que

divulguem e democratizem a utilização do método científico sejam fortemente incentivadas e compartilhadas.

Nesse sentido, o *NetLogo* se apresenta como uma ferramenta acessível e com uma abordagem diferenciada em relação a outros ambientes de modelagem pela utilização de agentes como elementos do modelo, sendo especialmente interessante no contexto da introdução de estudantes à modelagem de sistemas.

No entanto, durante o desenvolvimento verificou-se que o modelo criado pelo método das diferenças finitas foi executado de maneira mais lenta no *NetLogo* do que em outras ferramentas de modelagem computacional quando executado no mesmo *hardware*. Não é possível afirmar a partir dos dados de tempo de execução se isso se deve a uma não otimização do modelo em si ou a uma característica inerente ao *NetLogo*.

5.1 SUGESTÕES PARA TRABALHOS FUTUROS

Em relação aos produtos deste trabalho, recomenda-se que sejam feitos experimentos com uma metodologia aprimorada para o experimento físico, em uma faixa de temperatura maior e se possível usando sensores ópticos de temperatura como forma de corroborar ou refutar os resultados obtidos. Nota-se também a possibilidade de realizar um estudo de correlação por meio da coleta de uma grande quantidade de pontos de temperatura.

Como forma de verificar os benefícios de aprendizagem promovidos pelo método de Modelagem Bifocal recomenda-se a realização de um estudo que avalie esse quesito nos estudantes de ensino superior quando introduzidos a esse método, principalmente em relação ao desenvolvimento de um pensamento crítico sobre o experimento e de formulação de hipóteses sobre o problema de pesquisa.

O modelo desenvolvido pode ser aprimorado com a adaptação para um ambiente de modelagem 3D, o que certamente expandiria seu leque de aplicações, assim como o fator de correção desenvolvido no experimento computacional pode ser aprimorado como forma de melhor representar a perda de calor para o ambiente.

6 REFERÊNCIAS

BASTOS, Leonardo de Souza; SILVA, José Guilherme Santos da. Modelagem numérica do comportamento estrutural dinâmico e análise de conforto humano de edifícios altos submetidos à ação de cargas de vento não determinísticas. In: IBERIAN LATIN-AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGINEERING, 37., 2016, Brasília. **Anais...** Brasília: UnB, 2016. p. 1 - 19. Disponível em: <<http://periodicos.unb.br/index.php/ripe/article/view/23486>>. Acesso em: 29 mar. 2018.

BELHOT, Renato Vairo; OLIVEIRA NETO, José Dutra de. A solução de problemas no ensino de engenharia. In: SIMPÓSIO DE ENGENHARIA DE PRODUÇÃO, 13., 2006, Bauru. **Anais...** Bauru: Unesp, 2006. Disponível em: <http://www.simpep.feb.unesp.br/anais/anais_13/artigos/11.pdf>. Acesso em: 29 out. 2017.

BERMAN, Barry. 3-D printing: The new industrial revolution. **Business Horizons**, v. 55, n. 2, p.155-162, mar. 2012. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0007681311001790>>. Acesso em: 17 mar. 2018.

BLIKSTEIN, Paulo *et al.* Bifocal modeling: mixing real and virtual labs for advanced science learning. In: International Conference on Interaction Design and Children, 11., 2012, Bremen [Alemanha]. **Proceedings...** New York: ACM, 2012. p. 296 - 299. Disponível em: <<https://tltl.stanford.edu/sites/default/files/files/documents/publications/2012.IDC-BFGS.Bifocal.pdf>>. Acesso em: 25 out. 2017.

_____, Paulo. Bifocal Modeling: a study on the learning outcomes of comparing physical and computational models linked in real time. In: INTERNATIONAL CONFERENCE ON MULTIMODAL INTERACTION, 14., 2012, Santa Monica [Estados Unidos]. **Proceedings...** Santa Monica: ACM, 2012. p. 1 - 8. Disponível em: <<https://tltl.stanford.edu/publications/papers-or-book-chapters/bifocal-modeling-study-learning-outcomes>>. Acesso em: 02 maio 2018.

_____, Paulo. Digital Fabrication and 'Making' in Education: The Democratization of Invention. In: WALTER-HERRMANN, Julia; BÜCHING, Corinne (Ed.). **FabLab: Of Machines, Makers, and Inventors (Cultural and Media Studies)**. Bielefeld: Transcript Publishers, 2013. p. 203-224. Disponível em: <<https://tltl.stanford.edu/sites/default/files/files/documents/publications/2013.Book-B.Digital.pdf>>. Acesso em: 18 nov. 2017.

_____, Paulo; RAND, William; WILENSKY, Uri. Examining Group Behavior and Collaboration using ABM and Robots. In: AGENT, 2007., 2007, Evanston, IL. **Proceedings...** Evanston, IL: Agent, 2007. p. 1 - 12. Disponível em: <<https://ccl.northwestern.edu/2007/Blikstein-Rand-Wilensky Agent 2007 Paper v7.pdf>>. Acesso em: 05 mar. 2018.

_____, Paulo; WILENSKY, Uri. Bifocal modeling: a framework for combining computer modeling, robotics and real-world sensing. In: American Educational Research Association, 2007, Chicago. **Proceedings...** Chicago: Aera, 2007. p. 1 - 16. Disponível em: <https://ccl.northwestern.edu/papers/2007/09-bifocal_modeling.pdf>. Acesso em: 29 out. 2017.

_____, Paulo; WILENSKY, Uri. **NetLogo MaterialSim Grain Growth model**. 2005. Disponível em:
<<http://ccl.northwestern.edu/NetLogo/models/MaterialSimGrainGrowth>>. Acesso em: 08 maio 2018.

CARVALHO, Duílio Donizete Teixeira de. **Metodologia de Análise do Desempenho da Usina de Concentração da Samarco Mineração S.A. Baseada em Simulação das Operações**. 2003. 121 f. Dissertação (Mestrado) - Curso de Engenharia Mineral, Departamento de Engenharia de Minas, Universidade Federal de Ouro Preto, Ouro Preto, 2003. Disponível em:
<<http://www.repositorio.ufop.br/handle/123456789/2366>>. Acesso em: 19 mar. 2018.

CASALE, Adriana. **Aprendizagem Baseada em Problemas: desenvolvimento de competências para o ensino da engenharia**. 2013. 162 f. Tese (Doutorado) - Curso de Engenharia de Produção, Departamento de Engenharia de Produção, Universidade de São Paulo, São Carlos, 2013. Disponível em:
<<http://teses.usp.br/teses/disponiveis/18/18157/tde-12092013-085201>>. Acesso em: 14 mar. 2018.

CAVALCANTI, Paulo César F. H. et al. Modelagem e Simulação Computacional de Mecanismos. In: CONGRESSO NACIONAL DE ESTUDANTES DE ENGENHARIA MECÂNICA, 12., 2005, Ilha Solteira. **Anais...** Ilha Solteira: Abcm, 2005. p. 1 - 7. Disponível em:
<http://www.abcm.org.br/anais/creem/2005/pdf/trabalhos_completos/ee08.pdf>. Acesso em: 29 mar. 2018.

ÇENGEL, Yunus A.; GHAJAR, Afshin J. **Transferência de Calor e Massa: Uma abordagem prática**. 4. ed. New York [Estados Unidos]: McGraw-Hill, 2012.

CNPQ. **TABELA DE ÁREAS DO CONHECIMENTO**. Brasília: Ministério da Ciência, Tecnologia, Inovações e Comunicações, 2013. 22 p. Disponível em:
<<http://www.cnpq.br/documents/10157/186158/TabeladeAreasdoConhecimento.pdf>>. Acesso em: 26 jan. 2018.

FERNANDES, Lucas E. R. et al. A Rational Agent Controlling an Autonomous Vehicle: Implementation and Formal Verification. In: FORMAL VERIFICATION OF AUTONOMOUS VEHICLES WORKSHOP, 2017, Turim [Itália]. **Proceedings...** Turim: FVAV, 2017. p. 35 - 42. Disponível em: <<https://arxiv.org/abs/1709.02557>>. Acesso em: 02 maio 2018.

FUJIMOTO, Richard et al (Ed.). **Research Challenges in Modeling and Simulation for Engineering Complex Systems**. Arlington [Estados Unidos]: Springer International Publishing, 2016. 147 p. Disponível em:
<<http://www.trainingsystems.org/publications/Research-Challenges-in-Modeling-and-Simulation-for-Engineering-Complex-Systems.pdf>>. Acesso em: 05 abr. 2018.

GARCIA, Claudio. **Modelagem e Simulação de Processos Industriais e de Sistemas Eletromecânicos**. 2. ed. São Paulo: Editora da Universidade de São Paulo, 2013.

GIL, Antônio Carlos. **Como Elaborar Projetos de Pesquisa**. 4. ed. São Paulo: Editora Atlas S.a., 2002. 176 p.

GUILLERMO, Oscar Eduardo Patrón. **Uso de laboratórios virtuais de aprendizagem em mecânica dos fluídos e hidráulica na engenharia**. 2016. 160 f. Tese (Doutorado) - Curso de Programa de Pós-graduação em Informática na

Educação, Centro de Estudos Interdisciplinares em Novas Tecnologias da Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2016. Disponível em: <<http://hdl.handle.net/10183/149111>>. Acesso em: 28 mar. 2018.

HB. **TEC1-12715 Datasheet**. 2018. Disponível em: <<http://www.hebeiltd.com.cn/peltier.datasheet/TEC1-12715.pdf>>. Acesso em: 25 out. 2018.

HOSSEINPOUR, F.; HAJIHOSSEINI, H. Importance of Simulation in Manufacturing. **International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering**. [s. L.], p. 229-233. mar. 2009. Disponível em: <<https://waset.org/publications/8133/importance-of-simulation-in-manufacturing>>. Acesso em: 29 mar. 2018.

INCROPERA, Frank P.; DEWITT, David P. **Fundamentos de transferência de calor e de massa**. 4. ed. Rio de Janeiro: Livros Técnicos e Científicos, 1998. 494 p.

KELTON, W David; SADOWSKI, Randall P; ZUPICK, Nancy B. **Simulation with Arena**. 6. ed. New York [Estados Unidos]: Mcgraw-hill, 2015. 635 p.

LAW, Averill M.; KELTON, W. David. **Simulation Modeling and Analysis**. 3. ed. New York [Estados Unidos]: Mcgraw-hill, 2000. 760 p.

LITE. **NetLogo: Bifocal Modeling - Univali + Stanford**. 2017. Disponível em: <<https://github.com/UNIVALI-LITE/NetLogoProjects>>. Acesso em: 20 nov. 2017.

MACHADO, Walnice Brandão; LUZ, Talita Ribeiro da; PAIVA, Kely Cesar Martins de. Formação de competências na graduação em engenharia: estudo com docentes de uma instituição de ensino superior privada localizada na cidade de belo horizonte. **Trabalho & Educação**, Belo Horizonte, v. 2, n. 24, p.113-121, maio/jun. 2015. Bimestral.

MALISKA, Clovis Raimundo. **Transferência de calor e mecânica dos fluidos computacional**. 2. ed. Rio de Janeiro: Livros Técnicos e Científicos, 2004. 453 p.

MATHWORKS. **MATLAB**. 2018. Disponível em: <<https://www.mathworks.com/products/MATLAB.html>>. Acesso em: 20 abr. 2018.

MATWEB. **304 Stainless Steel**. 2018. Disponível em: <<http://www.matweb.com/search/datasheetText.aspx?bassnum=MQ304A>>. Acesso em: 20 out. 2018.

MELO, Kelly Jacqueline Moura de. **Aplicação do método das diferenças finitas explícito na solução da equação do calor para o caso transiente e unidimensional**. 2011. 46 f. TCC (Graduação) - Curso de Ciência e Tecnologia, Universidade Federal Rural do Semi-Árido, Angicos, 2011. Disponível em: <[http://www2.ufersa.edu.br/portal/view/uploads/setores/232/arquivos/Kelly Jacqueline Moura de Melo.pdf](http://www2.ufersa.edu.br/portal/view/uploads/setores/232/arquivos/Kelly%20Jacqueline%20Moura%20de%20Melo.pdf)>. Acesso em: 18 maio 2018.

NASA. Governo dos Estados Unidos. **Design Software Transforms How Commercial Jetliners Are Built**. 2018. Disponível em: <https://spinoff.nasa.gov/Spinoff2018/t_2.html>. Acesso em: 05 abr. 2018.

PALMER, Tim. Modelling: Build imprecise supercomputers. **Nature**, [s.l.], v. 526, n. 7571, p.32-33, 29 set. 2015. Disponível em: <https://www.nature.com/polopoly_fs/1.18437!/menu/main/topColumns/topLeftColumn/pdf/526032a.pdf>. Acesso em: 05 abr. 2018.

PEDRO, Annelize da Costa. **Diagnóstico da Infraestrutura de Materiais do Laboratório de Pesquisa e Tecnologia em Engenharia da UNIVALI**. 2013. 163 f. TCC (Graduação) - Curso de Engenharia Industrial Mecânica, Centro de Ciências Tecnológicas da Terra e do Mar, Universidade do Vale do Itajaí, Itajaí, 2013.

PEPPLOW, Luiz Amilton; SILVA, Jonny Carlos da; SANCHES, Pablo. Abordagem web para orientar a modelagem e simulação dinâmica unidimensional de sistemas de engenharia. In: Congresso Nacional de Engenharia Mecânica, 6., 2010, Campina Grande. **Anais...** Campina Grande: Abcm, 2010. p. 1 - 10.

PERINGER, Petr. **SIMLIB/C++**: SIMulation LIBrary for C++. 2017. Disponível em: <<http://www.fit.vutbr.cz/~peringer/SIMLIB/>>. Acesso em: 20 abr. 2018.

PILLEPICH, Annalisa et al. Simulating galaxy formation with the IllustrisTNG model. **Monthly Notices of The Royal Astronomical Society**, v. 473, n. 3, p.4077-4106, 12 out. 2017. Oxford University Press (OUP).
<<http://dx.doi.org/10.1093/mnras/stx2656>>. Disponível em:
<<https://academic.oup.com/mnras/article/473/3/4077/4494369>>. Acesso em: 29 mar. 2018.

QUEIROZ, Geraldo Nunes de; LIMA, Daniele dos Santos; CERQUEIRA, Daniel Lucas. A “PEDAGOGIA ESPAÇO-TEMPO” NO ENSINO DA ENGENHARIA: Experienciar o Objetivo (Output) e assimilar as Ferramentas Básicas (Input). In: CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA, 45., 2017, Joinville. **Anais...** Joinville: ABENGE, 2017. p. 1 - 10. Disponível em: <http://www.abenge.org.br/sis_submetidos.php?acao=abrir&codigo=COBENGE17_0001_00000314.pdf>. Acesso em: 17 mar. 2018.

ROSA, Juliana Aozane da; ARAÚJO, Maria Cristina Pansera de. As discussões sobre o ensino de física no congresso brasileiro de educação em engenharia. In: XXI JORNADA DE PESQUISA, 2., 2016, Ijuí. **Anais...** Ijuí: Unijuí, 2016. p. 1 - 5. Disponível em: <<https://www.publicacoeseventos.unijui.edu.br/index.php/salaconhecimento/article/viewFile/7005/5771>>. Acesso em: 18 nov. 2017.

RUSSELL, Stuart; NORVIG, Peter. **Inteligência Artificial**. 3. ed. Rio de Janeiro: Elsevier, 2013.

SCILAB. **About Scilab**. 2017. Disponível em: <<https://www.scilab.org/en/scilab/about>>. Acesso em: 20 abr. 2018.

SGORLON, Andrew B.; CRUZ, Jeferson Auto da. DESENVOLVIMENTO DE DISPOSITIVO PARA APLICAÇÃO DE MODELAGEM BIFOCAL: PLANO INCLINADO. In: CONGRESSO DE INICIAÇÃO CIENTÍFICA E TECNOLÓGICA DO IFSP, 7., 2016, Matão. **Anais...** Matão: Ifsp, 2016. p. 1 - 4. Disponível em: <<http://mto.ifsp.edu.br/images/CPI/Anais/IC/2343.pdf>>. Acesso em: 07 mar. 2018.

SILVA NETO, Gustavo Cunha da; VALDA, Luiz Henrique Carneiro; LOPES, Arlindo Pires. O método das diferenças finitas aplicado a problemas de transmissão de calor em regime transiente. In: Iberian-Latin-American Congress on Computational Methods in Engineering, 30., 2009, Buzios. **Anais...** Buzios: CILAMCE, 2009. p. 1 - 15. Disponível em: <https://www.researchgate.net/publication/262379080_O_METODO_DAS_DIFERENCAS_FINITAS_APLICADO_A_PROBLEMAS_DE_TRANSMISSAO_DE_CALOR_E_M_REGIME_TRANSIENTE>. Acesso em: 18 maio 2018.

SILVA, José Guilherme Santos da; VELLASCO, Pedro Colmar G. da S.; OLIVEIRA, Marcel Isandro Ribeiro de. A Importância da Modelagem de Estruturas Reticuladas Tridimensionais no Ensino de Engenharia. In: Congresso Brasileiro de Ensino De Engenharia, 2003, Rio de Janeiro. **Anais...** Rio de Janeiro: UERJ, 2003.

SOARES, Luiz Fernando Gomes. **Modelagem e simulação discreta de sistemas**. Rio de Janeiro: Campus, 1992. 250 p.

STREVELER, Ruth A. *et al.* Learning Conceptual Knowledge in the Engineering Sciences: Overview and Future Research Directions. **Journal of Engineering Education**, [s.l.], v. 97, n. 3, p.279-294, jul. 2008.

SULLIVAN, Randy. **Thermal conductivity of metals V2 Table**. 2018. Disponível em: <<https://chemdemos.uoregon.edu/demos/Which-metal-heats-up-fastest-Aluminum-Copper-or-Silver>>. Acesso em: 04 dez. 2018.

THERMTEST. **Thermal Conductivity Testing of Metals at Temperatures up to 200 °C**. 2018. Disponível em: <<https://thermtest.com/thermal-conductivity-testing-of-metals-at-temperatures-up-to-200-oc>>. Acesso em: 04 dez. 2018.

TISUE, Seth; WILENSKY, Uri. *NetLogo: A Simple Environment for Modeling Complexity*. In: INTERNATIONAL CONFERENCE ON COMPLEX SYSTEMS, .., 2004, Boston [Estados Unidos]. **Proceedings...** Boston: NECSI, 2004. p. 1 - 10. Disponível em: <<http://www.ccl.sesp.northwestern.edu/papers/NetLogo-iccs2004.pdf>>. Acesso em: 19 abr. 2018.

UNIVALI. **Engenharia Mecânica: Disciplinas**. 2018. Disponível em: <<https://www.univali.br/graduacao/engenharia-mecanica-itajai/disciplinas/Paginas/default.aspx>>. Acesso em: 22 mar. 2018.

VAZQUEZ, Mariana; MORABITO, Reinaldo; MARCONDES, Cesar. Caracterização, modelagem e simulação de enlace congestionado de uma universidade. **Gestão & Produção**. São Carlos, 2015. Disponível em: <<http://ref.scielo.org/g4wp7g>>. Acesso em: 25 abr. 2018.

VIEIRA, António et al. Comparison of SIMIO and ARENA simulation tools. In: INDUSTRIAL SIMULATION CONFERENCE, 2014, Skövde [Suécia]. **Proceedings...** 2014: EUROSIS, 2014. p. 1 - 9. Disponível em: <<http://hdl.handle.net/1822/36949>>. Acesso em: 20 abr. 2018.

WILENSKY, Uri. **NetLogo GasLab Adiabatic Piston model**. 1997. Disponível em: <<http://ccl.northwestern.edu/NetLogo/models/GasLabAdiabaticPiston>>. Acesso em: 08 maio 2018.

_____, Uri. **NetLogo Heat Diffusion model**. 1998. Disponível em: <<http://ccl.northwestern.edu/NetLogo/models/HeatDiffusion>>. Acesso em: 08 maio 2018.

_____, Uri. **NetLogo Polymer Dynamics model**. 2005. Disponível em: <<http://ccl.northwestern.edu/NetLogo/models/PolymerDynamics>>. Acesso em: 08 maio 2018.

_____, Uri. **NetLogo User Manual**. [s. L.]: Northwestern University, 2018. Disponível em: <<https://ccl.northwestern.edu/NetLogo/docs/NetLogo%20User%20Manual.pdf>>. Acesso em: 24 abr. 2018.

_____, Uri. **NetLogo GasLab Two Gas model**. Evanston [Estados Unidos]: Northwestern University, 1997. Disponível em: <<http://ccl.northwestern.edu/netlogo/models/GasLabTwoGas>>. Acesso em: 25 abr. 2018.

_____, Uri. **NetLogo**. Center for Connected Learning and Computer-Based Modeling, Northwestern University. 1999. Disponível em: <<https://ccl.northwestern.edu/NetLogo/>>. Acesso em: 14 nov. 2017.

_____, Uri. **Turbulence**. 2003. Disponível em: <<http://ccl.northwestern.edu/NetLogo/models/Turbulence>>. Acesso em: 08 maio 2018.

_____, Uri; RAND, William. **An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo**. Cambridge [Estados Unidos]: MIT Press, 2015.

XAVIER, Danilo Domingos. **Software livre na educação**. 2011. 44 f. Monografia (Especialização) - Curso de Informática da Educação, Instituto de Computação, Universidade Federal de Mato Grosso, Cuiabá, 2011. Disponível em: <<http://www.ic.ufmt.br/sites/default/files/field/pdf/Monografia/DaniloXavier.pdf>>. Acesso em: 26 out. 2017.

YAMANOOR, Narasimha Saii; YAMANOOR, Srihari. High Quality, Low Cost Education with the Raspberry Pi. In: GLOBAL HUMANITARIAN TECHNOLOGY CONFERENCE, 2017, San Jose [Estados Unidos]. **Proceedings...** San Jose: IEEE, 2017. p. 1 - 5. Disponível em: <<http://ieeexplore.ieee.org/document/8239274/>>. Acesso em: 17 mar. 2018.

ZHOU, Yufen; CHEN, Suren. Dynamic Simulation of a Long-Span Bridge-Traffic System Subjected to Combined Service and Extreme Loads. **Journal Of Structural Engineering**, [s.l.], v. 141, n. 9, p.1-18, set. 2015. American Society of Civil Engineers (ASCE). Disponível em: <[https://ascelibrary.org/doi/abs/10.1061/\(ASCE\)ST.1943-541X.0001188](https://ascelibrary.org/doi/abs/10.1061/(ASCE)ST.1943-541X.0001188)>. Acesso em: 29 mar. 2018.

APÊNDICES

APÊNDICE A. Código desenvolvido para o Arduino Uno utilizado no experimento de condução de calor.

// Disponível em <https://github.com/juliopw/TICT-Modelagem-Bifocal-NetLogo>

// Biblioteca da interface da tela LCD <http://blog.mklec.com/how-to-use-iici2c-serial-interface-module-for-1602-lcd-display>

// Com base na biblioteca de: <https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads> GNU General Public License, version 3 (GPL-3.0)

/*

Conexões:

SCL = A5

SDA = A4

VCC = 5V

GND = GND

*/

// LCD

#include < Wire.h > #include < LCD.h > #include < LiquidCrystal_I2C.h >

// Biblioteca estatística <http://playground.arduino.cc/Main/Statistics>

#include "Statistic.h"

// Criação de variáveis das bibliotecas

LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7); // 0x27 is the I2C bus address for an unmodified module

Statistic dadosMensuradosA;

Statistic dadosMensuradosB;

Statistic dadosMensuradosC;

Statistic dadosMensuradosD;

Statistic dadosMensuradosE;

Statistic dadosMensuradosFria;

Statistic dadosMensuradosQuente;

Statistic dadosMensuradosGeral;

// Termopares

#include < SPI.h >

//Inicialização de variáveis

double tempA, tempB, tempC, tempD, tempE, tempFria, tempQuente;

int identificadorDaMedida = 1;


```

int contadorNumMedidas = 0;
int intervaloMedidas;

void setup() {
  // Serial
  Serial.begin(115200);

  Serial.println(" | Experimento de Condução de Calor - TICT de Julio Patron Witwytzkyj |");
  Serial.println(" | Curso de Engenharia Mecânica - Universidade do Vale do Itajaí |\\n");

  Serial.println("Inicializando interface SPI e MAX6675... ");

  // Espera pela conexão serial
  while (!Serial) {
    delay(1);
  }

  Serial.println("Inicializando led indicador de medida... ");
  pinMode(LED_BUILTIN, OUTPUT);

  Serial.println("Inicializando biblioteca estatística... \\n");
  dadosMensuradosA.clear(); // Limpa valores estatísticos

  // Termopares
  SPI.begin();
  pinMode(7, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(2, OUTPUT);
  pinMode(8, OUTPUT);
  digitalWrite(7, HIGH);
  digitalWrite(6, HIGH);
  digitalWrite(5, HIGH);
  digitalWrite(4, HIGH);
  digitalWrite(3, HIGH);
  digitalWrite(2, HIGH);
  digitalWrite(8, HIGH);

  // LCD
  lcd.setBacklightPin(3, POSITIVE);
  lcd.setBacklight(HIGH); // NOTE: You can turn the backlight off by setting it to LOW instead of HIGH

```

```

lcd.begin(16, 2);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Aguardando");
lcd.setCursor(0, 1);
lcd.print("Serial");

recebeNumMedidas();
recebeIntervaloMedidas();

Serial.println("\nNumero da medida | A | B | C | D | E | Fria | Quente");

}

void loop() {
    tempA = readCelsius(8);
    dadosMensuradosA.add(tempA); // Adiciona medida aos dados estatísticos do ponto A
    dadosMensuradosGeral.add(tempA); // Adiciona medida aos dados estatísticos gerais

    tempB = readCelsius(2);
    dadosMensuradosB.add(tempB); // Adiciona medida aos dados estatísticos do ponto B
    dadosMensuradosGeral.add(tempB); // Adiciona medida aos dados estatísticos gerais

    tempC = readCelsius(3);
    dadosMensuradosC.add(tempC); // Adiciona medida aos dados estatísticos do ponto C
    dadosMensuradosGeral.add(tempC); // Adiciona medida aos dados estatísticos gerais

    tempD = readCelsius(4);
    dadosMensuradosD.add(tempD); // Adiciona medida aos dados estatísticos
    dadosMensuradosGeral.add(tempD); // Adiciona medida aos dados estatísticos gerais

    tempE = readCelsius(5);
    dadosMensuradosE.add(tempE); // Adiciona medida aos dados estatísticos
    dadosMensuradosGeral.add(tempE); // Adiciona medida aos dados estatísticos gerais

    tempQuente = readCelsius(6);
    dadosMensuradosQuente.add(tempQuente); // Adiciona medida aos dados estatísticos
    dadosMensuradosGeral.add(tempQuente); // Adiciona medida aos dados estatísticos gerais

    tempFria = readCelsius(7);
    dadosMensuradosFria.add(tempFria); // Adiciona medida aos dados estatísticos
    dadosMensuradosGeral.add(tempFria); // Adiciona medida aos dados estatísticos gerais

```

```

// Serial
Serial.print(String("    ") + identificadorDaMedida + String("    | "));
Serial.print(tempA + String(" | "));
Serial.print(tempB + String(" | "));
Serial.print(tempC + String(" | "));
Serial.print(tempD + String(" | "));
Serial.print(tempE + String(" | "));
Serial.print(tempQuente + String(" | "));
Serial.println(tempFria) + String("\n");

// LCD
lcd.clear();
lcd.setCursor(2, 0);
lcd.print("Medida " + identificadorDaMedida);
lcd.setCursor(0, 1);
lcd.print("A: " + String(tempA));
delay(intervaloMedidas / 4);

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("B: " + String(tempB));
lcd.setCursor(0, 1);
lcd.print("C: " + String(tempC));
delay(intervaloMedidas / 4);

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("D: " + String(tempD));
lcd.setCursor(0, 1);
lcd.print("E: " + String(tempE));
delay(intervaloMedidas / 4);

lcd.clear();
// fonte quente
lcd.setCursor(2, 0);
lcd.print("Quente: " + String(tempQuente));
// Fonte fria
lcd.setCursor(4, 1);
lcd.print("Fria: " + String(tempFria));
delay(intervaloMedidas / 4);

identificadorDaMedida++;

```

```

if (dadosMensuradosA.count() >= contadorNumMedidas) {
    estatistica();
    recebeNumMedidas();
    recebelIntervaloMedidas();
}
}

// Aguarda até que o valor do número de medidas desejado seja enviado pela porta serial.
void recebeNumMedidas() {
    Serial.print("Envie o número de medidas desejado... ");
    lcd.clear();
    contadorNumMedidas = 0;
    while (contadorNumMedidas == 0) { // Espera até que um valor válido de medidas seja recebido
        while (!Serial.available()) {

            // fonte quente
            lcd.setCursor(1, 0);
            lcd.print("Quente: " + String(readCelsius(7)));
            // Fonte fria
            lcd.setCursor(3, 1);
            lcd.print("Fria: " + String(readCelsius(6)));
            delay(100);
        } // Espera pelo comando da porta serial
        contadorNumMedidas = Serial.parseInt(); // Recebe o número de medidas
    }

    Serial.println(contadorNumMedidas);
}

void recebelIntervaloMedidas() {
    Serial.print("Envie o intervalo entre medidas (ms)... ");

    intervaloMedidas = 0;
    while (intervaloMedidas == 0) { // Espera até que um valor válido intervalo seja recebido
        while (!Serial.available()) {} // Espera pelo comando da porta serial
        intervaloMedidas = Serial.parseInt(); // Recebe o intervalo
    }

    Serial.println(intervaloMedidas);
}

// Função para leitura de termopares. Fonte: https://arduino.stackexchange.com/questions/37193/multiple-3-wire-spi-sensor-interfacing-with-arduino

```

```

double readCelsius(uint8_t cs) {
    uint16_t v;

    digitalWrite(cs, LOW);
    delay(20);
    v = SPI.transfer(0x00);
    v <<= 8;
    v |= SPI.transfer(0x00);
    digitalWrite(cs, HIGH);
    delay(20);

    if (v & 0x4) {
        // Termopar não acoplado
        return NAN;
    }

    v >>= 3;

    return v * 0.25;
}

void estatistica() {
    {
        // refazer string

        // A
        Serial.print("\nResultados: \n");
        Serial.print(String("A | " + String(dadosMensuradosA.average(), 2) + " | ")); // Média, 2 casas
        decimais
        Serial.print((dadosMensuradosA.pop_stdev(), 2) + " | "); // Desvio padrão, 2 casas decimais
        Serial.println(dadosMensuradosA.count() + "\n"); // Num medidas
        dadosMensuradosA.clear();

        // B
        Serial.print(String("B | " + String(dadosMensuradosB.average(), 2) + " | ")); // Média, 2 casas
        decimais
        Serial.print((dadosMensuradosB.pop_stdev(), 2) + " | "); // Desvio padrão, 2 casas decimais
        Serial.println(dadosMensuradosB.count() + "\n"); // Num medidas
        dadosMensuradosB.clear();

        // C
        Serial.print(String("C | " + String(dadosMensuradosC.average(), 2) + " | ")); // Média, 2 casas
        decimais
        Serial.print((dadosMensuradosC.pop_stdev(), 2) + " | "); // Desvio padrão, 2 casas decimais
    }
}

```

```

Serial.println(dadosMensuradosC.count() + "\n"); // Num medidas
dadosMensuradosC.clear();

// D
Serial.print(String("D | " + String(dadosMensuradosD.average(), 2) + " | ")); // Média, 2 casas
decimais
Serial.print((dadosMensuradosD.pop_stddev(), 2) + " | "); // Desvio padrão, 2 casas decimais
Serial.println(dadosMensuradosD.count() + "\n"); // Num medidas
dadosMensuradosD.clear();

// E
Serial.print(String("E | " + String(dadosMensuradosE.average(), 2) + " | ")); // Média, 2 casas
decimais
Serial.print((dadosMensuradosE.pop_stddev(), 2) + " | "); // Desvio padrão, 2 casas decimais
Serial.println(dadosMensuradosE.count() + "\n"); // Num medidas
dadosMensuradosE.clear();

// Fria
Serial.print(String("Fria | " + String(dadosMensuradosFria.average(), 2) + " | ")); // Média, 2 casas
decimais
Serial.print((dadosMensuradosFria.pop_stddev(), 2) + " | "); // Desvio padrão, 2 casas decimais
Serial.println(dadosMensuradosFria.count() + "\n"); // Num medidas
dadosMensuradosFria.clear();

// Quente
Serial.print(String("Quente | " + String(dadosMensuradosQuente.average(), 2) + " | ")); // Média,
2 casas decimais
Serial.print((dadosMensuradosQuente.pop_stddev(), 2) + " | "); // Desvio padrão, 2 casas decimais
Serial.println(dadosMensuradosQuente.count() + "\n\n"); // Num medidas
dadosMensuradosQuente.clear();

delay(3000);
recebeNumMedidas();

}

}

```

APÊNDICE B. Lista de partes utilizadas no experimento de condução de calor.

Tabela 21 – Lista de partes do experimento de condução de calor

Item	Quantidade
Arduino Uno	1 und.
MAX 6675	7 und.
Termopar tipo K - 1 mm	7 und.
Cabo USB tipo B	1 und.
Tela LCD 16x2	1 und.
Espuma de polietileno – 3 mm	0,03 m ²
Parafusos de fixação	5 und.
Fita dupla face	20 cm
Cabos diversos	1 m
Coluna de poliestireno	0,000096 m ³
Base de feltro	4 und.
Chapa de aço inoxidável – 2 mm	0,01 m ²

APÊNDICE C. Código do modelo computacional do experimento de condução de calor.

;Disponível em <https://github.com/juliopw/TICT-Modelagem-Bifocal-NetLogo>

```
extensions[bitmap]
```

```
;importar variáveis globais
```

```
globals [
```

```
  temperatura-media
```

```
  temperatura-media-anterior
```

```
  temperatura-maxima
```

```
  temperatura-minima
```

```
  temperatura-A
```

```
  temperatura-B
```

```
  temperatura-C
```

```
  temperatura-D
```

```
  temperatura-E
```

```
  cor_minima
```

```
  cor_maxima
```

```
  contador-medidas
```

```
]
```

```
breed [ pedacosdechapa pedacochapa ]
```

```
breed [ pontos-frios ponto-frio ]
```

```
breed [ pontos-quentes ponto- quente ]
```

```
breed [ sensores sensor ]
```

```
pedacosdechapa-own [
```

```
  temperatura
```

```
  temperatura-acima
```

```
  temperatura-abaixo
```

```
  temperatura-esquerda
```

```
  temperatura-direita
```

```
]
```



```

pontos-quentes-own [
  temperatura
]

```

```

pontos-frios-own [
  temperatura
]

```

```

sensores-own [
  temperatura
  nome
]

```

```

to setup
  clear-all
  reset-ticks

```

```

set cor_minima 12
set cor_maxima 18

```

```

set-patch-size tamanho-do-elemento
resize-world -280 / tamanho-do-elemento 260 / tamanho-do-elemento -260 / tamanho-do-elemento 260 /
tamanho-do-elemento

```

```

ask patches [set pcolor white]

```

```

if imagem-amostra = "Chapa 1" [
  import-pcolors "chapa.png"
]

```

```

if imagem-amostra = "Chapa 2" [
  import-pcolors "chapa2.png"
]

```

```

if imagem-amostra = "Chapa 3" [
  import-pcolors "chapa3.png"
]

```

```

if imagem-amostra = "Amostra TICT" [
  import-pcolors "Amostra TICT.png"
]

```

```

if imagem-amostra = "Amostra TICT com Peltier" [
  import-pcolors "Amostra TICT com Peltier.png"
]

```

;conversão de cores

```

ask patches [
  if (pcolor < 1 and pcolor >= 0) or (pcolor < 11 and pcolor >= 10) or (pcolor < 21 and pcolor >= 20) or
  (pcolor < 41 and pcolor >= 40) or (pcolor < 51 and pcolor >= 50) or (pcolor < 61 and pcolor >= 60) or
  (pcolor < 61 and pcolor >= 60) or (pcolor < 71 and pcolor >= 70) or (pcolor < 81 and pcolor >= 80) or (pcolor
  < 91 and pcolor >= 90) or (pcolor < 101 and pcolor >= 100) [ set pcolor 0 ];preto

  if pcolor >= 1 and pcolor < 10 [ set pcolor 9.9 ] ;branco

  if (pcolor >= 11 and pcolor < 19) or (pcolor >= 21 and pcolor < 29) or (pcolor >= 41 and pcolor < 49) [ set
  pcolor 14.9 ] ;quente

  if (pcolor >= 71 and pcolor < 79) or (pcolor >= 81 and pcolor < 89) or (pcolor >= 91 and pcolor < 99) or
  (pcolor >= 101 and pcolor < 109) [ set pcolor 95.1 ];frio
]

```

criar-chapa

```

ask turtles [
  set temperatura temperatura-inicial
]

```

aplica-temperatura-nas-fontes

```

ask pedacosdechapa [
  colorir
]

```

end

to go

aplica-temperatura-nas-fontes

```

if mouse-down? [
  ask pedacosdechapa with [round xcor = round mouse-xcor and round ycor = round mouse-ycor] [
    set temperatura temperatura-fonte-quente
    ask pedacosdechapa-on neighbors4 [
      set temperatura temperatura-fonte-quente
    ]
  ]
]

```

calcula-temperatura

```

if abs (temperatura-media - temperatura-media-anterior) <= 0.00005 [
  escrever-temperaturas
  set contador-medidas contador-medidas + 1
  ask pedacosdechapa [set temperatura temperatura-inicial]
]
set temperatura-media-anterior temperatura-media

```

```

if contador-medidas >= 1 [ stop ]

```

tick

```

;wait 0.01

```

end

to aplica-temperatura-nas-fontes

```

ifelse fonte-quente = true [
  ask pontos-quentes [
    set temperatura temperatura-fonte-quente
  ]
][
  ask pontos-quentes [
    if any? turtles-on neighbors4 [
      set temperatura mean [temperatura] of turtles-on neighbors4
    ]
  ]
]

```

```

ifelse fonte-fria = true [
  ask pontos-frios [
    set temperatura temperatura-fonte-fria
  ]
][
  ask pontos-frios [
    if any? turtles-on neighbors4 [
      set temperatura mean [temperatura] of turtles-on neighbors4
    ]
  ]
]

```

end

to criar-chapa

```

ask patches [
  if pcolor = black and (not any? turtles-here) [
    sprout-pedacosdechapa 1 [
      set color black

      if formato-do-elemento = "Círculo" [
        set shape "circle"
        set size 1
      ]

      if formato-do-elemento = "Quadrado" [
        set shape "square"
        set size 1
      ]
    ]
  ]
]

```

```

if pcolor = 14.9 and (not any? turtles-here) [
  sprout-pontos-quentes 1 [
    set color red
  ]
]

```

```

    if formato-do-elemento = "Círculo" [
        set shape "circle"
        set size 1
    ]

    if formato-do-elemento = "Quadrado" [
        set shape "square"
        set size 1
    ]
]

if pcolor = 95.1 and (not any? turtles-here) [
    sprout-pontos-frios 1 [
        set color blue

        if formato-do-elemento = "Círculo" [
            set shape "circle"
            set size 1
        ]

        if formato-do-elemento = "Quadrado" [
            set shape "square"
            set size 1
        ]
    ]
]
end

to calcula-temperatura

ask sensores with [nome = "A"] [
    set temperatura mean [temperatura] of turtles-here
    set temperatura-A temperatura
]

ask sensores with [nome = "B"] [

```

```

    set temperatura mean [temperatura] of turtles-here
    set temperatura-B temperatura
]

ask sensores with [nome = "C"] [
    set temperatura mean [temperatura] of turtles-here
    set temperatura-C temperatura
]

ask sensores with [nome = "D"] [
    set temperatura mean [temperatura] of turtles-here
    set temperatura-D temperatura
]

ask sensores with [nome = "E"] [
    set temperatura mean [temperatura] of turtles-here
    set temperatura-E temperatura
]

ask pedacosdechapa [
    if any? turtles-on neighbors4 and pcolor != red and pcolor != blue[

        ;cálculo da temperatura
        set temperatura mean [temperatura] of turtles-on neighbors4

        ;fator de correção
        ifelse temperatura > temperatura-media [
            set temperatura (temperatura - fator-de-correcao * temperatura / 100)
        ] [
            set temperatura (temperatura + fator-de-correcao * temperatura / 100)
        ]

        colorir
    ]
]

```

```

set temperatura-maxima max [temperatura] of pedacosdechapa
set temperatura-minima min [temperatura] of pedacosdechapa

```

```

set temperatura-media mean [temperatura] of pedacosdechapa

```

```

end

```

```

to inserir-sensores

```

```

  let coordenada-minima-x min [xcor] of pedacosdechapa
  let coordenada-maxima-x max [xcor] of pedacosdechapa
  let coordenada-minima-y min [ycor] of pedacosdechapa
  let coordenada-maxima-y max [ycor] of pedacosdechapa

```

```

create-sensores 1 [
  set xcor round (coordenada-minima-x - (coordenada-minima-x * 2 / 5))
  set ycor round (coordenada-maxima-y - (coordenada-maxima-y * 2 / 5))
  set shape "A"
  set nome "A"
  set size world-width / 30
]

```

```

create-sensores 1 [
  set xcor round ((coordenada-maxima-x - (coordenada-maxima-x * 2 / 5)))
  set ycor round (coordenada-maxima-y - (coordenada-maxima-y * 2 / 5))
  set shape "B"
  set nome "B"
  set size world-width / 30
]

```

```

create-sensores 1 [
  set xcor 0
  set ycor 0
  set shape "C"
  set nome "C"
  set size world-width / 30
]

```

```

create-sensores 1 [

```

```

set xcor round ((coordenada-minima-x - (coordenada-minima-x * 2 / 5))
set ycor round (coordenada-minima-y - (coordenada-minima-y * 2 / 5))
set shape "D"
set nome "D"
set size world-width / 30
]

```

```

create-sensores 1 [
  set xcor round ((coordenada-maxima-x - (coordenada-maxima-x * 2 / 5))
  set ycor round (coordenada-minima-y - (coordenada-minima-y * 2 / 5))
  set shape "E"
  set nome "E"
  set size world-width / 30
]

```

end

to escrever-temperaturas

```

file-open ( word temperatura-fonte-quente " Q " temperatura-fonte-fria " F.txt")
file-print (word temperatura-A " ; " temperatura-B " ; " temperatura-C " ; " temperatura-D " ; "
temperatura-E " ; " temperatura-fonte-fria " ; " temperatura-fonte-quente "\r\n" )
file-close
end

```

to colorir

```

if temperatura != 0 and cores = "Contínua (Tons de vermelho)" [
  set color abs ((cor_minima - cor_maxima) * ((temperatura - temperatura-maxima) / (temperatura-
minima - temperatura-maxima)) + cor_maxima)
]

if temperatura != 0 and cores = "Contínua (Branco - Vermelho - Preto)" [
  set color scale-color red temperatura temperatura-minima temperatura-maxima
]

if temperatura != 0 and cores = "10 Cores graduadas (Vermelho - Amarelo - Verde - Azul)" [
  temperatura-graduada1
]

if temperatura != 0 and cores = "10 Cores graduadas (Amarelo - Vermelho - Azul)" [

```



```

    temperatura-graduada2
]

if temperatura != 0 and cores = "20 Cores graduadas (Vermelho - Amarelo - Verde - Azul)" [
    temperatura-graduada3
]

if temperatura != 0 and cores = "Tons de cinza" [
    temperatura-tons-de-cinza
]

if temperatura != 0 and cores = "Cores desordenadas (apenas linhas isotérmicas)" [
    set color temperatura
]

end

to temperatura-graduada1

if ( temperatura >= temperatura-minima ) and ( temperatura <= 0.1 * temperatura-maxima ) [
    ;set color scale-color blue temperatura ( 0.2 * temperatura-maxima ) ( 0.4 * temperatura-maxima )
    set color 105
]

if ( temperatura > 0.1 * temperatura-maxima ) and ( temperatura <= 0.2 * temperatura-maxima ) [
    set color 95
]

if ( temperatura > 0.2 * temperatura-maxima ) and ( temperatura <= 0.3 * temperatura-maxima ) [
    set color 85
]

if ( temperatura > 0.3 * temperatura-maxima ) and ( temperatura <= 0.4 * temperatura-maxima ) [
    set color 75
]

if ( temperatura > 0.4 * temperatura-maxima ) and ( temperatura <= 0.5 * temperatura-maxima ) [
    set color 65
]

```

```

if ( temperatura > 0.5 * temperatura-maxima ) and ( temperatura <= 0.6 * temperatura-maxima ) [
    set color 55
]

```

```

if ( temperatura > 0.6 * temperatura-maxima ) and ( temperatura <= 0.7 * temperatura-maxima ) [
    set color 45
]

```

```

if ( temperatura > 0.7 * temperatura-maxima ) and ( temperatura <= 0.8 * temperatura-maxima ) [
    set color 25
]

```

```

if ( temperatura > 0.8 * temperatura-maxima ) and ( temperatura <= 0.9 * temperatura-maxima ) [
    set color 14
]

```

```

if ( temperatura > 0.9 * temperatura-maxima ) and ( temperatura <= 1.0 * temperatura-maxima ) [
    set color 15
]

```

end

to temperatura-graduada2

```

if ( temperatura >= temperatura-minima ) and ( temperatura <= 0.1 * temperatura-maxima ) [
    set color 103
]

```

```

if ( temperatura > 0.1 * temperatura-maxima ) and ( temperatura <= 0.2 * temperatura-maxima ) [
    set color 105
]

```

```

if ( temperatura > 0.2 * temperatura-maxima ) and ( temperatura <= 0.3 * temperatura-maxima ) [
    set color 95
]

```

```

if ( temperatura > 0.3 * temperatura-maxima ) and ( temperatura <= 0.4 * temperatura-maxima ) [

```

```

    set color 85
]

if ( temperatura > 0.4 * temperatura-maxima ) and ( temperatura <= 0.5 * temperatura-maxima ) [
    set color 16
]

if ( temperatura > 0.5 * temperatura-maxima ) and ( temperatura <= 0.6 * temperatura-maxima ) [
    set color 15
]

if ( temperatura > 0.6 * temperatura-maxima ) and ( temperatura <= 0.7 * temperatura-maxima ) [
    set color 25
]

if ( temperatura > 0.7 * temperatura-maxima ) and ( temperatura <= 0.8 * temperatura-maxima ) [
    set color 44
]

if ( temperatura > 0.8 * temperatura-maxima ) and ( temperatura <= 0.9 * temperatura-maxima ) [
    set color 47
]

if ( temperatura > 0.9 * temperatura-maxima ) and ( temperatura <= 1.0 * temperatura-maxima ) [
    set color 49
]

end

to temperatura-graduada3

if ( temperatura >= temperatura-minima ) and ( temperatura <= 0.05 * temperatura-maxima ) [
    set color 103
]

if ( temperatura > 0.05 * temperatura-maxima ) and ( temperatura <= 0.1 * temperatura-maxima ) [
    set color 105
]

```

```
if ( temperatura > 0.1 * temperatura-maxima ) and ( temperatura <= 0.15 * temperatura-maxima ) [  
    set color 93  
]
```

```
if ( temperatura > 0.15 * temperatura-maxima ) and ( temperatura <= 0.2 * temperatura-maxima ) [  
    set color 95  
]
```

```
if ( temperatura > 0.2 * temperatura-maxima ) and ( temperatura <= 0.25 * temperatura-maxima ) [  
    set color 83  
]
```

```
if ( temperatura > 0.25 * temperatura-maxima ) and ( temperatura <= 0.3 * temperatura-maxima ) [  
    set color 85  
]
```

```
if ( temperatura > 0.3 * temperatura-maxima ) and ( temperatura <= 0.35 * temperatura-maxima ) [  
    set color 73  
]
```

```
if ( temperatura > 0.35 * temperatura-maxima ) and ( temperatura <= 0.4 * temperatura-maxima ) [  
    set color 75  
]
```

```
if ( temperatura > 0.4 * temperatura-maxima ) and ( temperatura <= 0.45 * temperatura-maxima ) [  
    set color 63  
]
```

```
if ( temperatura > 0.45 * temperatura-maxima ) and ( temperatura <= 0.5 * temperatura-maxima ) [  
    set color 65  
]
```

```
if ( temperatura > 0.5 * temperatura-maxima ) and ( temperatura <= 0.55 * temperatura-maxima ) [  
    set color 53  
]
```

```
if ( temperatura > 0.55 * temperatura-maxima ) and ( temperatura <= 0.6 * temperatura-maxima ) [  
    set color 53  
]
```

```
    set color 55
]

if ( temperatura > 0.6 * temperatura-maxima ) and ( temperatura <= 0.65 * temperatura-maxima ) [
    set color 43
]

if ( temperatura > 0.65 * temperatura-maxima ) and ( temperatura <= 0.7 * temperatura-maxima ) [
    set color 45
]

if ( temperatura > 0.7 * temperatura-maxima ) and ( temperatura <= 0.75 * temperatura-maxima ) [
    set color 47
]

if ( temperatura > 0.75 * temperatura-maxima ) and ( temperatura <= 0.8 * temperatura-maxima ) [
    set color 23
]

if ( temperatura > 0.8 * temperatura-maxima ) and ( temperatura <= 0.85 * temperatura-maxima ) [
    set color 25
]

if ( temperatura > 0.85 * temperatura-maxima ) and ( temperatura <= 0.9 * temperatura-maxima ) [
    set color 13
]

if ( temperatura > 0.9 * temperatura-maxima ) and ( temperatura <= 0.95 * temperatura-maxima ) [
    set color 15
]

if ( temperatura > 0.95 * temperatura-maxima ) and ( temperatura <= 1.0 * temperatura-maxima ) [
    set color 17
]

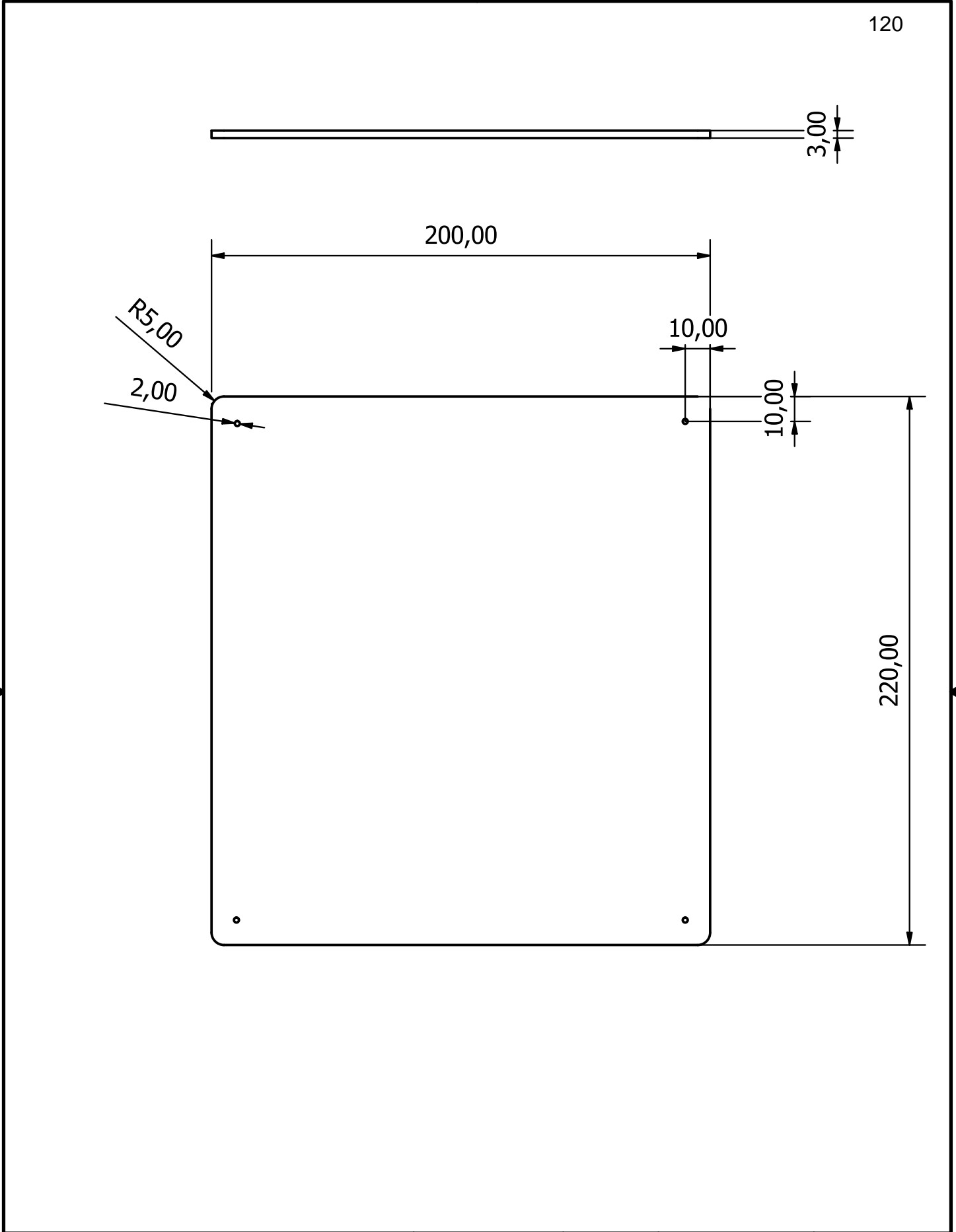
end

to temperatura-tons-de-cinza
```

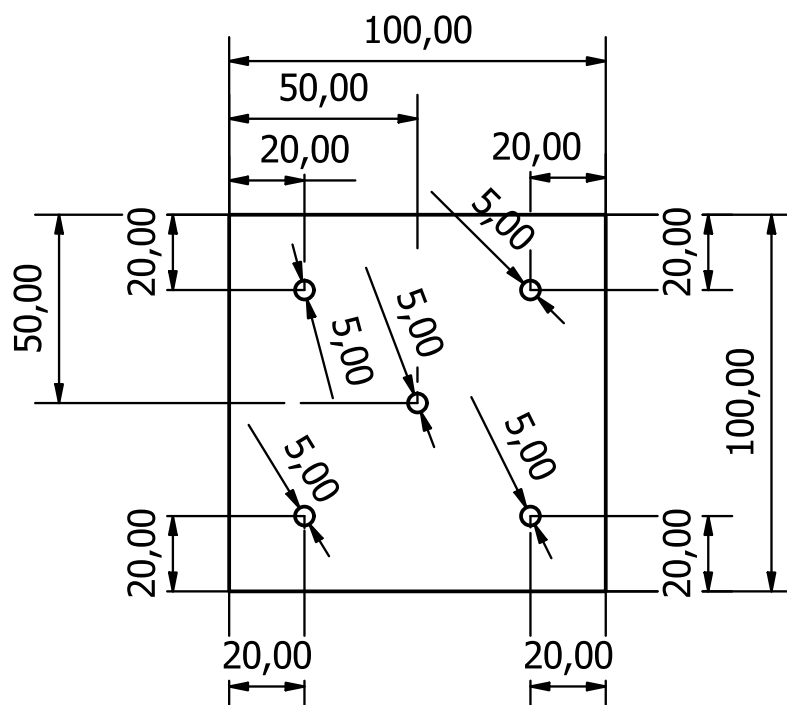
```
if ( temperatura >= temperatura-minima ) and ( temperatura <= 0.2 * temperatura-maxima ) [  
    set color 2  
]  
  
if ( temperatura > 0.2 * temperatura-maxima ) and ( temperatura <= 0.4 * temperatura-maxima ) [  
    set color 4  
]  
  
if ( temperatura > 0.4 * temperatura-maxima ) and ( temperatura <= 0.6 * temperatura-maxima ) [  
    set color 6  
]  
  
if ( temperatura > 0.6 * temperatura-maxima ) and ( temperatura <= 0.8 * temperatura-maxima ) [  
    set color 8  
]  
  
if ( temperatura > 0.8 * temperatura-maxima ) and ( temperatura <= 1 * temperatura-maxima ) [  
    set color 9.9  
]  
end
```

APÊNDICE D. Projeto do experimento de condução de calor.

Nas próximas páginas são apresentados os desenhos técnicos usados para corte das chapas de MDF usadas na construção do experimento de condução de calor.



Projetado por Julio Patron Witwytzkj	Data 26/08/2018	Escala 1:2	Quantidade 1	Unidade mm
Universidade do Vale do Itajai	Experimento_Conducao_de_Calor			
	Peca_inferior			Folha 2 / 6



Projetado por

Julio Patron Witwytzkyj

Data

26/08/2018

Escala

1:2

Quantidade

1

Unidade

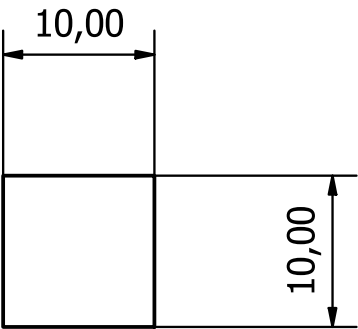
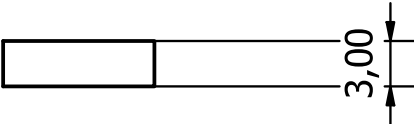
mm

Universidade do Vale do Itajai

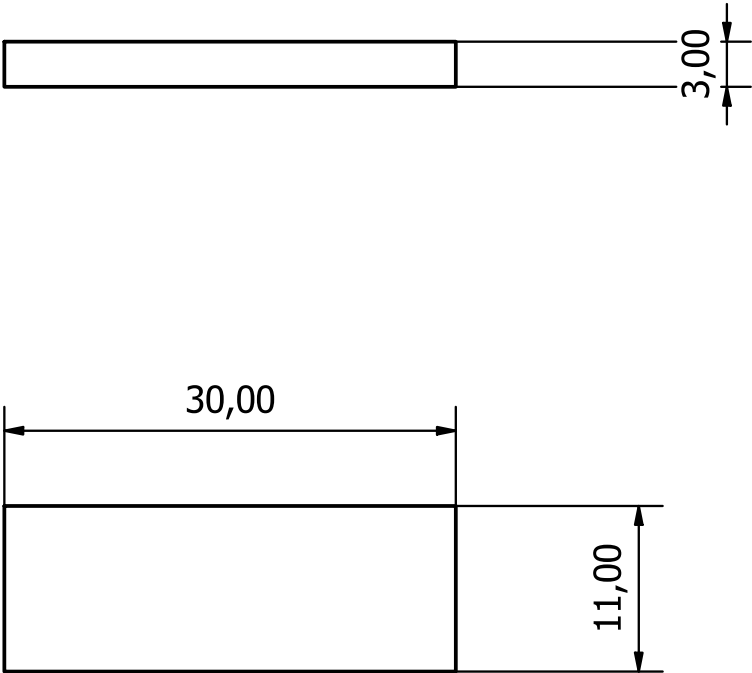
Experimento_Conducao_de_Calor

Fixador_da_amostra

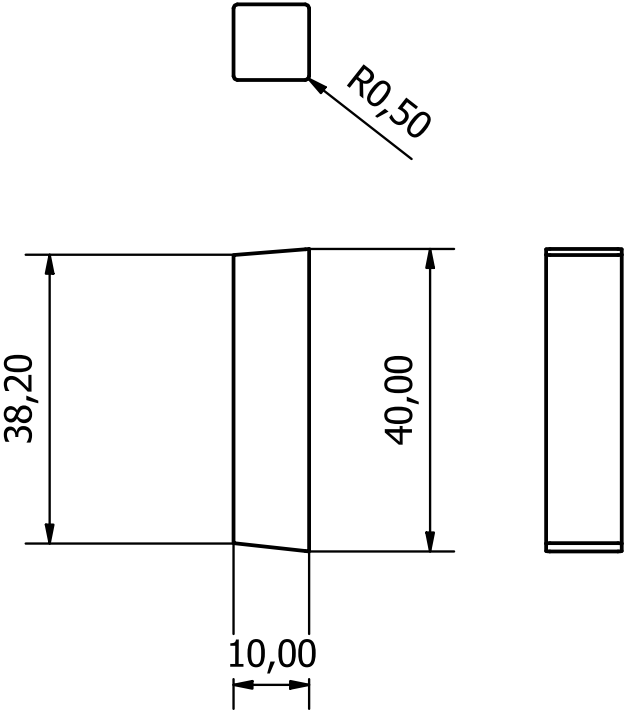
Folha
3 / 6



Projetado por Julio Patron Witwytzkj	Data 26/08/2018	Escala 2:1	Quantidade 8	Unidade mm
Universidade do Vale do Itajai		Experimento_Conducao_de_Calor		
		Apoio_da_amostra		Folha 4 / 6



Projetado por Julio Patron Witwytzkyj	Data 26/08/2018	Escala 2:1	Quantidade 3	Unidade mm
Universidade do Vale do Itajai	Experimento_Conducao_de_Calor			
	Apoio_da_fonte_fria		Folha 5 / 6	



Projetado por Julio Patron Witwytzkyj	Data 26/08/2018	Escala 1:1	Quantidade 2	Unidade mm
Universidade do Vale do Itajai	Experimento_Conducao_de_Calor			
	Coluna			Folha 6 / 6

ANEXOS

ANEXO A. Tabela de condutividade térmica

Tabela 22 – Condutividade térmica de materiais à temperatura ambiente

Material	k, W/m.K	Material	k, W/m.K
Diamante	2.300	Água (l)	0,607
Prata	429	Pele humana	0,37
Cobre	401	Madeira (carvalho)	0,17
Ouro	317	Hélio (g)	0,152
Alumínio	237	Borracha macia	0,13
Ferro	80,2	Fibra de vidro	0,043
Mercúrio (l)	8,54	Ar (g)	0,026
Vidro	0,78	Uretano, espuma rígida	0,026
Tijolo	0,72		

Fonte: Adaptado de Çengel e Ghajar (2012)