

Anais da

# XXXV Semana Acadêmica da Matemática 2021



Anais da

# XXXV Semana Acadêmica da Matemática

03/11/2021 a 09/11/2021

Comissão organizadora da XXXV Semana Acadêmica da Matemática:

Fabiana Magda Garcia Papani (Coordenadora)  
Cintya Akemi Okawa  
Clezio Aparecido Braga  
Dulcyene Maria Ribeiro  
Felipe Leandro da Silva Costa  
Fernanda Guerra  
Jean Sebastian Toillier  
Marieli Vanessa Rediske de Almeida  
Pâmela Gonçalves

Comitê científico:

Alessandra dos Santos  
Andre Vicente  
Andreia Büttner Ciani  
Arlen Elise Sella Langer  
Clezio Aparecido Braga  
Daniela Maria Grande Vicente  
Dulcyene Maria Ribeiro  
Fabiana Magda Garcia Papani  
Felipe Leandro da Silva Costa  
Flavio Roberto Dias Silva

Jean Sebastian Toillier  
Luciana Pagliosa Carvalho Guedes  
Marieli Vanessa Rediske de Almeida  
Pamela Gonçalves  
Paulo Domingos Conejo  
Rogerio Luis Rizzi  
Rosângela Villwock  
Simone Aparecida Miloca

Arte da Capa:

Clezio Aparecido Braga

Diagramação:

Clezio Aparecido Braga

Dados Internacionais de Catalogação-na-Publicação (CIP)

|       |                                                                                                                                                                  |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| S471a | Semana Acadêmica de Matemática (35.: 2021: Cascavel – PR)                                                                                                        |
|       | Anais da XXXV Semana Acadêmica de Matemática / Fabiana Magda Garcia Papani (Coordenadora). --- Cascavel (PR): UNIOESTE, 2021. 52p.                               |
|       | ISSN 2526-0804                                                                                                                                                   |
|       | Vários autores                                                                                                                                                   |
|       | Versão <i>on-line</i>                                                                                                                                            |
|       | Evento realizado no Campus de Cascavel no período de 03/11/2019 a 09/11/2021                                                                                     |
|       | 1. Matemática-Estudo e Ensino. 2. Ensino superior - Evento. I. Papani, Fabiana Magda Garcia (Coord.). II. Universidade Estadual do Oeste do Paraná. III. Título. |
|       | CDD 20.ed. 510.0711                                                                                                                                              |

Rosângela A A Silva CRB/9-1810

# Apresentação

A Semana Acadêmica da Matemática está na sua XXXV edição. Este é o evento de extensão mais tradicional promovido pelo Curso de Matemática, da UNIOESTE campus de Cascavel. É um evento com periodicidade anual.

Na programação da XXXV Semana Acadêmica de Matemática figuram palestras, minicursos e comunicações orais. As comunicações orais resultam da inscrição dos participantes na modalidade de apresentadores de trabalhos.

Nesta edição da Semana Acadêmica de Matemática, 7 trabalhos foram inscritos e aceitos para apresentação oral e publicação nos anais do evento. São em geral trabalhos resultados das pesquisas de Iniciação Científica e de Monografia desenvolvidos por alunos do curso de Matemática. Registramos também trabalhos realizados por professores do Curso de Matemática da UNIOESTE - Cascavel, e de alunos de outros cursos que desenvolveram suas pesquisas com teor matemático. A apresentação destes trabalhos no evento tem o objetivo de compartilhar os conhecimentos adquiridos pelos alunos e professores nos seus respectivos projetos. O registro destes trabalhos servirá para que os futuros alunos possam também fazer uso deste conhecimento.

A comissão organizadora agradece aos autores pelo envio dos trabalhos e também à comissão científica pelas contribuições dadas durante o processo de avaliação e correção dos trabalhos.

A comissão organizadora.

## Índice de Trabalhos

|                                                                                                    |    |
|----------------------------------------------------------------------------------------------------|----|
| Ensino de função exponencial a partir de uma situação problema.....                                | 4  |
| Exemplos numéricos de otimização quadrática irrestrita com Python                                  | 11 |
| Geometria Analítica em Motor de Jogos .....                                                        | 18 |
| O problema do caixeiro viajante.....                                                               | 28 |
| O uso de novas tecnologias no ensino remoto nas aulas de matemática:<br>relato de experiência..... | 33 |
| Otimização de formas quadráticas.....                                                              | 41 |
| Simplicidade dos grupos alternados .....                                                           | 46 |

## Ensino de função exponencial a partir de uma situação problema

Laís Dri da Rosa  
Universidade Estadual do Oeste do Paraná  
laisdridarosa@hotmail.com

**Resumo:** Esse trabalho baseia-se em uma situação problema proposta para o ensino de funções exponenciais. Visando a introdução da função exponencial em um contexto não rotineiro recorrendo a um fenômeno natural denominado avalanches e valendo-se de conhecimentos prévios demonstrados. A atividade proposta foi inspirada pelo projeto Matemática Multimídia da Universidade Estadual de Campinas, a fim de apresentar o conteúdo de maneira contextualizada e visual correlacionando com a aplicação cotidiana dos conceitos de função. Nessa perspectiva, podemos ressaltar que a matemática não é uma ciência isolada, está em constante troca com diversos aspectos da vida cotidiana e de outras áreas do conhecimento e esse ponto deve ser ressaltado nas práticas de ensino.

**Palavras-chave:** Função Exponencial; Ensino de Funções; Educação Matemática.

### 1 Introdução

Há décadas vem sendo realizados estudos e pesquisas na área de Educação Matemática buscando compreender o processo de ensino e aprendizagem da matemática nos diversos níveis de ensino. Diversas pesquisas resultaram em metodologias que proporcionam uma maior liberdade para estimular o trabalho em grupo, a utilização do raciocínio, solidificando elos entre conteúdos, oportunizando que os estudantes reconheçam diversas possibilidades de resolução e tracem seu próprio caminho em busca de seu aprendizado.

Sobre isso, os Parâmetros Curriculares Nacionais (PCN) apresentam como orientação que

Os alunos confrontados com situações problema, novas, mas compatíveis com os instrumentos que já possuem ou que possam adquirir no processo, aprendem a desenvolver estratégias de enfrentamento, planejando etapas, estabelecendo relações, ampliam sua autonomia e capacidade de comunicação e de argumentação (BRASIL, 1998, p. 266).

Por sua vez, a Base Nacional Comum Curricular (BNCC) traz como uma de suas competências específicas de matemática e suas tecnologias no Ensino Médio:

Utilizar estratégias, conceitos, definições e procedimentos matemáticos para interpretar, construir modelos e resolver problemas em diversos contextos, analisando a plausibilidade dos resultados e a adequação das soluções propostas, de modo a construir argumentação consistente (BRASIL, 2018, p. 531).

Ainda desses documentos norteadores podemos destacar o ensino de funções, o qual tem sua introdução nos anos finais do Ensino Fundamental, com a apresentação de conceitos iniciais, noções intuitivas, o reconhecimento em situações problemas, conceito de variável e relação entre duas variáveis. Para o Ensino Médio, pode-se destacar a relação dos conceitos de função com diversos contextos, o

aprofundamento dos conhecimentos prévios, a consolidação dos conceitos e a aplicação desses conhecimentos em diversas situações reais que podem ser observadas.

Nesse trabalho iremos destacar uma abordagem não rotineira para o ensino da função exponencial, o qual é previsto para o primeiro ano do Ensino Médio, buscando com isso estimular os estudantes a serem mais ativos em seus processos de aprendizagem, se sentirem desafiados a desenvolver suas habilidades e ampliar seus conhecimentos.

A partir desse ponto de vista, o professor se torna uma ferramenta de apoio durante esse processo, buscando compreender o raciocínio dos estudantes e guiá-los para superar suas dificuldades. De acordo com Polya,

O melhor é, porém, ajudar o estudante com naturalidade. O professor deve colocar-se no lugar do aluno, perceber o ponto de vista deste, procurar compreender o que se passa em sua cabeça e fazer uma pergunta ou indicar um passo que poderia ter ocorrido ao próprio estudante (POLYA, 1995, p. 1).

Podemos ressaltar que a matemática se encontra em constante troca com diversas áreas do conhecimento, especificamente os conceitos relacionados ao conteúdo de função, que pode ser correlacionado com as mais diversas situações cotidianas, a fim de ser contextualizado na prática de ensino. Isso é destacado pelo Parâmetros Curriculares Nacionais para o Ensino Médio (PCNEM):

Além das conexões internas à própria Matemática, o conceito de função desempenha também papel importante para descrever e estudar através da leitura, interpretação e construção de gráfico, o comportamento de certos fenômenos tanto do cotidiano, como de outras áreas do conhecimento, como a Física, Geografia ou Economia (BRASIL, 2000, p. 43).

Podemos afirmar, a partir dos documentos norteadores, que está prevista a correspondência entre conteúdos matemáticos com outras áreas do conhecimento, buscando uma maior contextualização dos conceitos matemáticos, a fim de uma otimização do aprendizado.

Com isso, a proposta de situação problema que foi apresentada aos estudantes do 1º ano do Ensino Médio, consistia em partir de um evento natural, para o qual deveria ser realizada uma simulação utilizando alguns materiais fornecidos, e por meio dos dados obtidos encontrar as relações e interpretá-las.

A reflexão sobre a utilização de situação problema para o trabalho com função exponencial deu-se por meio da observação do ensino de funções do 1º e 2º grau, as quais possuem contextualizações e situações problemas usualmente empregadas para as práticas de ensino, o que não é visto com tanta frequência no ensino de função exponencial. A partir disso, pretendeu-se buscar por uma atividade que propusesse uma situação problema por meio de um experimento prático, com o intuito de ser uma introdução interativa para o conteúdo.

## 2 Metodologia

Considerando a busca por um ambiente em que o estudante esteja mais ativo em seu processo de aprendizagem e baseado no experimento denominado *Avalanche*, proposto pelo projeto Matemática Multimídia da Universidade Estadual de Campinas – Unicamp, o qual aborda o comportamento de uma avalanche por meio da Função Logarítmica (OLIVEIRA E FERRAZ, 2017), aqui pretende-se apresentar uma interpretação e adaptação da proposta para o contexto buscado, relatando o plano de aula realizado e também a maneira que deu-se o desenvolvimento desse plano.

A Função Logarítmica está presente nos conteúdos programáticos para o primeiro ano do Ensino Médio, e para o seu ensino são necessários conceitos previamente apresentados, entre eles a equação exponencial, plano cartesiano, coordenadas cartesianas, variáveis, domínio e imagem.

Com base na sugestão de Oliveira e Ferraz (2017), que propõem modelar matematicamente avalanches por meio de grãos de pipoca, grãos de feijão e copos descartáveis, os estudantes, divididos em grupos, devem simular avalanches com os materiais fornecidos, observando a intensidade pela quantidade de grãos que desmoronam, amparados pelo roteiro fornecido com questões norteadoras que visam auxiliar no desenvolvimento do experimento. Os grupos devem preencher o copo com o máximo de grãos possível de maneira que obtenham um morrinho estável sobre o copo, após isso serão colocados os grãos um a um sobre o amontoado de grãos e realizada a contagem dos que caem aos colocá-los.

A partir dos dados coletados devem ser construídos gráficos, que em seguida serão interpretados seguindo o roteiro proposto. Alguns questionamentos apresentados se referem a se o gráfico obtido se assemelha a alguma função conhecida, que relação pode ser observada entre a intensidade e a quantidade, que informações podem ser obtidas por meio da tabela e do gráfico e a que conclusão é possível chegar com as informações obtidas.

## 3 Desenvolvimento

Neste experimento os estudantes deverão modelar matematicamente o fenômeno de avalanches, utilizando grãos de pipoca e copos descartáveis, no qual os estudantes irão simular avalanches com os materiais fornecidos, verificando as intensidades por meio da quantidade de grãos que desmoronam dos seus copos e, com as anotações, poderão construir gráficos que formam uma curva representativa do evento simulado. A partir disso, poderão linearizar os gráficos por meio de logaritmos, permitindo a análise da função que modela o fenômeno.

A atividade foi desenvolvida em dois momentos distintos, sendo o primeiro momento com estudantes do ensino superior, e um segundo momento com estudantes do primeiro ano do Ensino Médio, os quais não haviam estudado funções exponenciais até o momento.

O primeiro momento, em que foi realizada a atividade com estudantes de ensino superior, deu-se por uma atividade avaliativa da disciplina de Didática aplicada ao ensino de Matemática, na qual desenvolvemos atividades planejadas para Ensino Fundamental ou Ensino Médio da maneira que deveriam ser desenvolvidas com os estudantes, porém com a oportunidade de discutir sobre o desenvolvimento, realizar contribuições para aprimorar o plano de aula, além de compartilhar experiências, sendo elas vivenciadas como estudantes ou educadores.

Aqui nos focaremos no momento da aplicação da atividade com alunos do primeiro ano do Ensino Médio. Primeiramente seria entregue o material de apoio para os estudantes, separados em grupos. O material consistia nas instruções e questões norteadoras para o desenvolvimento da proposta, os exercícios de fixação envolvendo Função Logarítmica, uma tabela de logaritmo na base 2, bem como no material que usaríamos na realização do experimento (copos descartáveis, grãos de milho), seguido de alguns esclarecimentos acerca dos materiais e de que maneira deveriam proceder.

A Figura 1 exibe a folha de instruções apresentada aos alunos para a realização do procedimento e descreve as etapas envolvidas.

#### Procedimento

##### **Etapas 1: Coleta dos dados**

Preencha o recipiente com o grão que foi dado ao seu grupo. Coloque o máximo possível até que a estrutura fique estável.

Cuidado para não esbarrar na montagem!

Uma vez montado seu morrinho inicial, faça o seguinte:

1. Coloque vagarosamente um grão em qualquer ponto do morrinho;
2. Verifique se houve alguma avalanche: se não houver, coloque um novo grão, se houver, conte quantos grãos caíram e anote essa quantidade;
3. Repita esse procedimento por 10 minutos, nunca se esquecendo de anotar a intensidade de cada avalanche ocorrida.

Agora que você realizou as suas avalanches, construa uma tabela, relacionando quantas vezes ocorreram avalanches com cada intensidade, que corresponde a quantidade de grãos que caíram do morro.

##### **Etapas 2: Representação gráfica**

Construa uma tabela geral para cada tipo de grão. Faça o gráfico de  $I \times Q$ .

O gráfico obtido se parece com alguma função que você conhece? Qual? Por quê?

**Etapas 3: Uma nova representação** Na verdade, existem modelos matemáticos para analisar esse tipo de fenômeno. Em particular, o modelo usado para avalanches se torna mais fácil de compreender quando analisamos o gráfico de  $\log_2(I) \times \log_2(Q)$ .

Para obter esse gráfico:

1. Calcule, com a ajuda do anexo, o logaritmo de todos os valores das duas colunas da tabela que você montou, obtendo um nova tabela preenchida com  $\log_2(I)$  e  $\log_2(Q)$ .
2. Construa o gráfico de  $\log_2(I) \times \log_2(Q)$ . Trace com uma régua uma reta que se encaixe bem nos pontos do gráfico e calcule a sua equação.

Figura 1: Folha das instruções.

Fonte: Acervo da autora.

Inicialmente, deveria ser completo o copo descartável e com os grãos ao máximo obtendo uma estrutura estável, feito isso seriam colocados grãos, um a um, e observado se houve alguma avalanche, caso houvesse seria anotado a quantidade. O procedimento se repetiria por um período de 10 minutos. Com as informações obtidas constrói-se uma tabela relacionando a quantidade de avalanches (Q) com a intensidade (I - quantidade de grãos que caíram). A partir dessa tabela constrói-se o gráfico  $I \times Q$  e, observando o gráfico obtido, deveria ser verificado se possui características de uma função.



Figura 2: Desenvolvimento da atividade.

Fonte: Acervo da autora.

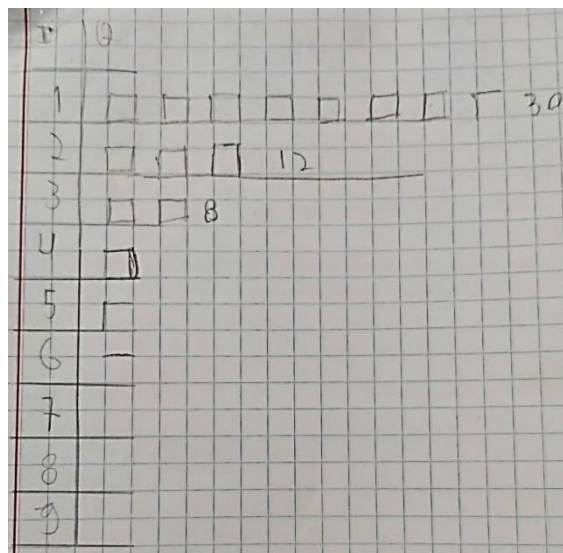


Figura 3: Quantidade de avalanches com a intensidade.

Fonte: Acervo da autora.

Após verificar se o gráfico representa uma função, propõe-se uma nova representação, sendo que existem dois modelos matemáticos para analisar esse tipo de fenômeno, porém o modelo utilizado para avalanches torna a compreensão mais simples, analisando o gráfico de  $\log_2(I) \times \log_2(Q)$ .

Para obter a equação da reta que procuramos deve ser realizada a manipulação de  $Q = \frac{a}{I^b}$ , utilizando as propriedades de logaritmos, sendo  $\log_2(Q) = \log_2(a) + \log_2(I^{-b})$ , do qual obtemos  $\log_2(Q) = \log_2(a) - b \cdot \log_2(I)$ , e se considerarmos  $\log_2(Q) = y$ ,  $\log_2(a) = c$  e  $\log_2(I) = x$ , obtemos então a equação da reta  $y = c - bx$ .

Então utilizando a tabela de logaritmos na base 2, reescreve-se a tabela obtida com o experimento, calculando-se o logaritmo de todos os valores das duas colunas, obtendo-se assim uma tabela com colunas  $\log_2(I)$  e  $\log_2(Q)$ . A partir dos pontos obtidos na nova tabela obtém-se o gráfico de uma reta.

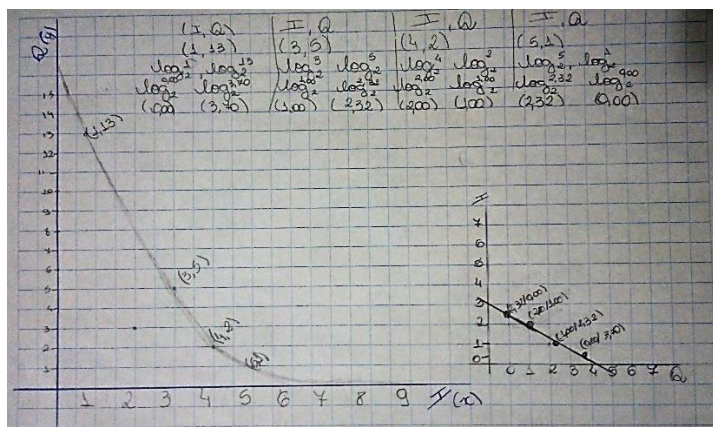


Figura 4: Resolução da atividade proposta.

Fonte: Acervo da autora.

Utilizando os pontos obtidos é possível calcular o coeficiente angular  $b = \frac{\Delta y}{\Delta x} = \frac{y_1 - y_0}{x_1 - x_0}$ , e substituindo na equação calculada anteriormente e utilizando  $y = 0$  para encontrar o coeficiente linear, consequentemente obtém-se a equação da reta representada no gráfico.

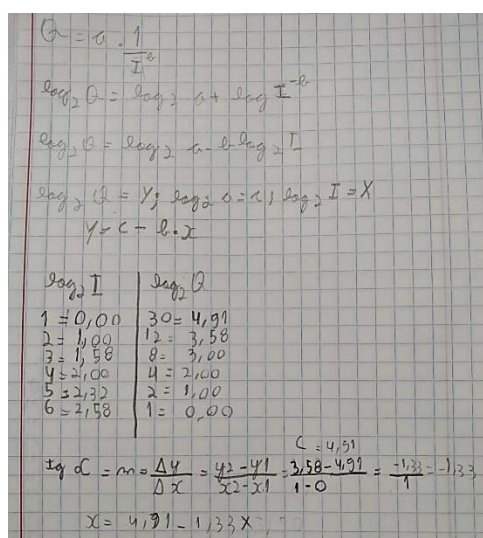


Figura 5: Resolução da atividade proposta.

Fonte: Acervo da autora.

Analisando todas as informações obtidas desde o início do problema proposto, obteve-se a interpretação de que avalanches de menor intensidades ocorrem com mais frequência do que avalanches de maior intensidade, podendo ser dito o mesmo sobre os desmoronamentos e até outros eventos que

estão presentes na natureza, tais como terremotos, incêndios naturais em florestas, tempestades solares, extinção de espécies biológicas, congestionamento em trânsito, entre outros.

### Conclusões

Com esse experimento foi possível mobilizar diversos conceitos matemáticos para a solução de uma situação problema, entre eles equações, coordenadas dependentes e independentes, plano cartesiano, coeficientes angular e linear, construção e interpretação de tabelas e gráficos.

Em ambos os momentos de desenvolvimento foi possível observar o comprometimento, concentração e cooperação na resolução do problema proposto, discutindo os métodos a serem utilizados e os resultados obtidos. Mesmo distribuindo tarefas, os estudantes ainda estavam ativos em todo o processo de resolução, contribuindo com ideias e interpretações da situação, porém os estudantes do Ensino Médio demonstraram um maior entusiasmo e curiosidade, pois não estão habituados a serem protagonistas de seu processo de aprendizagem. Acrescentando o fato de ser um conteúdo novo, mas que engloba diversos conceitos já conhecidos, o conteúdo se tornou mais interessante e motivador.

### Referências

- BRASIL. Ministério da Educação. Secretaria de Educação Média e Tecnológica. **Parâmetros Curriculares Nacionais**: ensino médio. Brasília, 1998. 364 p.
- BRASIL. Ministério da Educação. Secretaria de Educação Média e Tecnológica. **Base Nacional Comum Curricular**: ensino médio. Brasília, 2018.
- BRASIL, Secretaria de Educação Média e Tecnológica. **Parâmetros Curriculares Nacionais**: Ensino Médio (Parte III - Ciências da Natureza, Matemática e suas tecnologias). Brasília: 2000.
- OLIVEIRA, Samuel Rocha de; FERRAZ, Mariana Sacrini Ayres. **AVALANCHES**. <<http://m3.ime.unicamp.br/recursos/1366>>. Acesso em: 16/10/2016.
- PAIVA, Manoel. **Matemática Paiva**. 3º ano. 2. ed. São Paulo: Moderna, 2013.
- POLYA, George. **A arte de resolver problemas**: Um aspecto do método matemático. Rio de Janeiro: Interciência, 1995. 179 p. Tradução e adaptação: Heitor Lisboa de Araújo.

## Exemplos numéricos de otimização quadrática irrestrita com Python

Alex Augusto Nunes Machado  
Universidade Estadual do Oeste do Paraná - *Campus* Cascavel  
augusttoalex@gmail.com

Simone Aparecida Miloca  
Universidade Estadual do Oeste do Paraná - *Campus* Cascavel  
smiloca@gmail.com

**Resumo:** Este trabalho tem como tema central a programação quadrática irrestrita, que é um tipo de problema de Otimização não-linear. Problemas dessa natureza aparecem em diversas situações práticas, tanto na fase de modelagem, como em processos de busca de soluções em outros tipos de problemas de Otimização. Isto posto, este trabalho apresenta uma breve contextualização deste campo da Matemática bem como formaliza o que vem a ser um problema de programação quadrática. Na sequência, apresenta os métodos de busca de solução conhecidos como Método do Gradiente e Método de Newton. Por fim, apresenta os resultados obtidos com a implementação computacional, realizada via linguagem de programação Python, de tais métodos de busca aplicados a problemas de programação quadrática irrestrita, problemas estes que também foram gerados via Python.

**Palavras-chave:** Programação quadrática irrestrita definida positiva; Implementação computacional; Python.

## 1 Introdução

Conforme apontam Ribeiro e Karas (2013), direta ou indiretamente a noção de otimização faz parte do dia-a-dia do ser humano moderno, dado que as pessoas responsáveis pela tomada de decisão (em qualquer campo do conhecimento) deparam-se com problemas nos quais necessitam obter resultados não somente precisos mas também eficientes.

Nesse sentido, conforme pontuam Hillier e Lieberman (2013, p.1), desde o desenvolvimento da sociedade industrializada, e sobretudo com a eclosão da segunda guerra mundial na qual os militares britânicos e norte-americanos visando alocar os recursos da melhor maneira possível “(...) convocaram grande número de cientistas para aplicar uma abordagem científica para lidar com este e outros problemas táticos e estratégicos”, a demanda por ferramentas que auxiliem nesse processo decisório por parte de organizações, públicas ou privadas, nos mais diversos campos, sejam no âmbito científico ou empresarial, cresceu substancialmente. É neste contexto que a Otimização Matemática se origina.

Isto posto, o que vem a ser esse campo de estudo denominado *Otimização Matemática*? Visando responder a esse questionamento, deve-se observar que tal palavra tem sua origem derivada da palavra *otimizar* que, segundo o dicionário online Dicio, significa: ocasionar cir-

cunståncias mais proveitosas para, retirar o que há de melhor em, aprimorar, melhorar. (OTIMIZAR, 2021).

Porém, essa definição deixa a desejar quando se refere ao contexto matemático, sendo assim Ribeiro e Karas (2013) caracterizam a Otimização como encontrar pontos de mínimo ou máximo de uma função  $f: A \subset \mathbb{R}^n \rightarrow \mathbb{R}$  que está sujeita a determinadas restrições (outras funções). Tal função  $f$  é denominada *função objetivo* e emerge, quando pessoas responsáveis pela tomada de decisão deparam-se com problemas nos quais necessitam obter resultados precisos e, além disso,

(...) um único aspecto adequado de um problema pode ser isolado e caracterizado por um objetivo, seja lucro ou perda em um ambiente de negócios, velocidade ou distância em um problema físico, retorno esperado no ambiente de investimentos de risco (...) (LUENBERGER; YE, 2016, p. 1, tradução do autor)

Com isso, por meio de um processo de modelagem<sup>1</sup>, é possível traduzir a situação que demanda ser resolvida para o “mundo matemático”. Essa tradução, de modo geral, é representada por

$$\begin{array}{ll} \text{otimizar} & f(x) \\ \text{sujeito a} & x \in \Omega \subset \mathbb{R}^n \end{array} \quad (i)$$

no qual  $\Omega$  representa um conjunto de restrições que podem ser de igualdade ou desigualdade. Portanto, (i) formaliza, em termos matemáticos, o que vem a ser um problema de Otimização Matemática.

Assim, percebe-se então que o modelo consiste em uma espécie de representação do “mundo real” no “mundo matemático”. Esse processo de tradução como indicam Goldbarg e Luna (2005, p. 9) envolve “a percepção do elaborador do modelo (ou equipe de elaboração), uma faculdade cognitiva de alto nível. As *fórmulas* ou *equações* do modelo não existem prontas e acabadas na natureza, elas têm que ser identificadas ou criadas”. Todavia, embora tal processo vise a extração da essência do problema, conforme salientam Hillier e Lieberman (2013), vale destacar que “a correspondência entre experiência e modelo formal está longe de ser perfeita: a tradução está sujeita a erros, simplificações e falhas de comunicação.” (MARTÍNEZ; SANTOS, 1995, p. 1).

Em consequência disso, tanto o conjunto  $\Omega \subset \mathbb{R}^n$  quanto a função objetivo podem assumir diversos formatos, uma vez que sua obtenção está diretamente ligada ao processo de formulação do modelo matemático. Nesse sentido, conforme apontam Ribeiro e Karas (2013) e Goldbarg e Luna (2005) é possível agrupar os problemas de otimização em algumas categorias.

De modo geral, podemos dividir os problemas como sendo de otimização restrita e otimização irrestrita, isto é, problemas com e sem restrições. Cada uma dessas categorias pode ser dividida em outras duas, a saber problemas lineares e não lineares, os quais, por sua vez, podem ainda serem subdivididos em problemas de variáveis contínuas, discretas ou mistas.

<sup>1</sup>Para uma discussão um pouco mais detalhada sobre modelagem matemática recomenda-se a leitura do capítulo inicial de Goldbarg e Luna (2005).

Sendo assim, este trabalho versará sobre uma classe de problemas contido no conjunto dos problemas não lineares de variáveis contínuas, mais precisamente problemas de programação quadrática que de acordo com Hillier e Lieberman (2013) aceita que sua função objetivo possua lei de formação contendo termos quadráticos ou produto entre duas variáveis.

Em termos matemáticos, (ii) descreve esse modelo,

$$\begin{array}{ll} \text{minimizar} & f(x) = \frac{1}{2}x^T Ax + b^T x + c \\ \text{sujeito a} & x \in \mathbb{R}^n, \end{array} \quad (ii)$$

com  $b \in \mathbb{R}^n$ ,  $c \in \mathbb{R}$  e  $A \in M_{n \times n}(\mathbb{R})$  simétrica definida positiva.

O estudo dessa classe de problemas se justifica pelos seguintes motivos:

1. Qualquer problema do “mundo real”, independente do campo ou área do conhecimento no qual está inserido, que possa ser modelado sob as hipóteses da programação quadrática é passível de ser resolvido (HILLIER; LIEBERMAN, 2013);
2. Além de que “(...) uma metodologia comum para solucionar problemas de otimização genéricos linearmente restritos é resolver uma sequência de aproximações de programação quadrática.” (HILLIER; LIEBERMAN, 2013, p. 523). Isto é, a programação quadrática pode ser utilizada como uma estratégia de “aproximação” para problemas não quadráticos.

Na próxima seção descrevemos dois métodos de busca de solução para problemas de otimização irrestrita, mais especificamente o Método do Gradiente e o de Newton. Com isso, visando exemplificar alguns aspectos teóricos relacionados a convergência, bem como aspectos computacionais como o número de iterações e o tempo computacional para resolução de problemas quadráticos irrestritos, um banco de funções foi construído e utilizado em implementações efetuadas usando a linguagem de programação Python, sendo os resultados apresentados na última seção.

## 2 Algoritmos de busca

Em geral a obtenção de soluções para problemas de otimização é uma tarefa relativamente árdua, pois como visto na introdução, os problemas de otimização geralmente surgem por meio de um processo de modelagem de situações que se apresentam no “mundo real”, e com isso as funções (objetivo e/ou restrições) envolvidas podem não ser tão elementares, ou mesmo que sejam, podem demandar muitas variáveis, e assim, a solução direta pode não ser uma boa opção.

Com isso, entram em cena os *métodos de busca* que funcionam como um meio de, a partir de um palpite inicial  $x_0 \in \mathbb{R}^n$  para a solução do problema, encontrar outro ponto,  $x_1 \in \mathbb{R}^n$ , que pode ou não ser a solução ótima. Caso não seja, mas o novo ponto é *suficientemente* bom, pode-se optar por parar o processo. No entanto, se esse novo palpite não for uma boa solução aproximada, repete-se o procedimento de modo a obter um novo ponto de  $\mathbb{R}^n$  e então submetê-lo a nova análise. De modo geral, esse procedimento pode ser repetido por  $k$  vezes, caracterizando

assim um processo iterativo que, idealmente, conforme destacam Luenberger e Ye (2016) e Ribeiro e Karas (2013), geram uma sequência de pontos de  $\mathbb{R}^n$  que se aproximam da solução ótima.

Em geral, visando a obtenção de soluções aproximadas cada vez melhores, cada iteração desse processo iterativo consiste em escolher uma *direção de descida* ( $d_k$ ) e determinar qual será o *tamanho do passo* ( $t_k$ ), isto é, o quanto deve-se seguir na direção de descida associada a iteração  $k$ . Para isso, conforme colocam Ribeiro e Karas (2013), as escolhas de  $d_k$  e  $t_k$  não podem ser arbitrárias pois, se assim o fossem, poderiam gerar uma sequência de pontos que, embora possua pontos de acumulação estes podem não ser estacionários, isto é, podem não ser bons candidatos a minimizadores.

Diante disso, uma pergunta natural que emerge é: “*Como escolher  $d_k$  e  $t_k$  de modo conveniente ao propósito de minimizar  $f$  mais rapidamente?*” a qual pode ser parafraseada em: “*Como escolher as direções e as amplitudes de cada passo em direção ao mínimo de modo eficiente?*”.

Uma resposta satisfatória para esse questionamento depende de qual algoritmos de busca foi escolhido, todavia dentre os inúmeros que foram desenvolvidos e melhorados ao longo de décadas, descreveremos aqui o Método do Gradiente e o Método de Newton. No primeiro, a ideia central consiste em escolher  $-\nabla f(x_k)$  como sendo a direção de descida enquanto que a determinação do tamanho do passo é determinado ao minimizarmos a função de uma variável  $g$  tal que  $g(t_k) = f(x_k + t_k d_k)$ . Já, o Método de Newton<sup>2</sup> propõe tomar  $x_{k+1}$  como sendo o ponto que minimiza a aproximação de Taylor de segunda ordem na iteração  $k$ .

Quando aplicados a modelos de programação quadrática é possível mostrar que, no Método do Gradiente, a determinação do tamanho do passo é dada por

$$t_k = -\frac{\nabla f(x_k)^T d_k}{d_k^T A d_k} = \frac{\nabla f(x_k)^T \nabla f(x_k)}{\nabla f(x_k)^T A \nabla f(x_k)},$$

enquanto no Método de Newton a solução é obtida em apenas uma iteração. Para mais detalhes, ver Friedlander (1994) e Machado (2021).

### 3 Resultados e discussões

Com o objetivo de analisar algumas informações estudadas na parte teórica dos métodos apresentados, como por exemplo o número de iterações e tempo computacional, realizamos a implementação dos métodos de Newton e do gradiente para o caso particular de problemas de minimização quadrática irrestrita. Para tanto, o processo de testes numéricos realizado com auxílio da linguagem Python consistiu em:

1. Geração de matrizes definidas positivas, geração de palpites iniciais e geração de termos lineares;

---

<sup>2</sup>Para uma discussão teórica mais detalhada desse método sugere-se a leitura de Ribeiro e Karas (2013) e Martins e Pereira (2010).

2. Implementação dos métodos do gradiente e de Newton e levando em consideração as matrizes e vetores geradas no item anterior.

Tal implementação foi realizada em um *notebook* pessoal com processador Intel(R) Core(TM) i7-5500U munido de dois núcleos e quatro *thread(s)* possuindo velocidade mínima de 500 MHz e máxima de 3000 MHz. Além de contar com dois chips de memória RAM SO-DIMM DDR3 de 4096MB cada, operando a 1600 MHz, bem como um SSD GIGABYTE GP-GSTFS31240GNTD.

Diante disso, as Tabela 1 e 2 apresentam os resultados obtidos com a implementação do Método do Gradiente à problemas quadráticos e 7 e 1000 variáveis.

Tabela 1: Método do Gradiente: problema com 7 variáveis

| Número máximo de iterações | Aproximação para $  \nabla f(x_k)  $ | Quantidade de testes | Iterações utilizadas | Tempo médio de cada teste em segundos |
|----------------------------|--------------------------------------|----------------------|----------------------|---------------------------------------|
| 2000                       | $10^{-10}$                           | 10                   | 1394                 | 0,035                                 |
| 2000                       | $10^{-12}$                           | 10                   | 1620                 | 0,037                                 |

Fonte: Produzido pelo autor

Tabela 2: Método do Gradiente: problema com 1000 variáveis

| Número máximo de iterações | Aproximação para $  \nabla f(x_k)  $ | Quantidade de testes | Iterações utilizadas | Tempo médio de cada teste em segundos |
|----------------------------|--------------------------------------|----------------------|----------------------|---------------------------------------|
| 50000                      | $10^{-3}$                            | 10                   | 22461                | 57,164                                |
| 50000                      | $10^{-4}$                            | 10                   | 27039                | 66,956                                |
| 50000                      | $10^{-5}$                            | 10                   | 31617                | 77,282                                |
| 50000                      | $10^{-6}$                            | 10                   | 36195                | 88,004                                |
| 75000                      | $10^{-10}$                           | 10                   | 54794                | 131,249                               |

Fonte: Produzido pelo autor

Ou seja, utilizando o Método do Gradiente, o tempo para encontrar as aproximações desejadas aumentou significativamente se comparado aos problemas iniciais, porém considerando as limitações de *hardware*, ainda podemos dizer que o tempo ficou dentro de um limite aceitável.

Conforme estabelecido anteriormente, o método de Newton obtém a solução de problemas quadráticos em apenas uma iteração<sup>3</sup>. Em consequência, os testes realizados com problemas de 3, 5, 7, 150, 500 e 1000 variáveis apresentaram imensa diferença de desempenho temporal quando comparado com o método de Newton. A Tabela 3 apresenta os tempos médios de resolução associados a cada problema pelo método de Newton.

Sendo assim, visando descobrir qual a ordem dos problemas que poderiam ser resolvidos pelo método de Newton em tempo semelhante ao que o Método do Gradiente levou para solucionar o problema de 1000 variáveis com 10 casas decimais de precisão, foram realizados mais seis conjuntos de 10 testes, sendo 10 com 1500 variáveis, 10 com 2000, 10 com 2500, 10 com 3000, 10 com 3300 e 10 com 3350. Os dados obtidos estão dispostos na Tabela 4.

<sup>3</sup>Desde que a matriz A envolvida seja inversível.

Tabela 3: Método de Newton: dados relativos ao método de Newton

| Número de variáveis | Aproximação para $  \nabla f(x_k)  $ | Quantidade de testes | Tempo médio de cada teste em segundos |
|---------------------|--------------------------------------|----------------------|---------------------------------------|
| 3                   | $10^{-12}$                           | 10                   | 0,003                                 |
| 5                   | $10^{-13}$                           | 10                   | 0,003                                 |
| 7                   | $10^{-12}$                           | 10                   | 0,004                                 |
| 150                 | $10^{-10}$                           | 10                   | 0,039                                 |
| 500                 | $10^{-10}$                           | 10                   | 0,684                                 |
| 1000                | $10^{-9}$                            | 10                   | 4,335                                 |

Fonte: Produzido pelo autor

Tabela 4: Método de Newton: dados relativos ao método de Newton

| Número de variáveis | Aproximação para $  \nabla f(x_k)  $ | Quantidade de testes | Tempo médio de cada teste em segundos |
|---------------------|--------------------------------------|----------------------|---------------------------------------|
| 1500                | $10^{-9}$                            | 10                   | 11,211                                |
| 2000                | $10^{-9}$                            | 10                   | 27,688                                |
| 2500                | $10^{-8}$                            | 10                   | 52,794                                |
| 3000                | $10^{-9}$                            | 10                   | 93,523                                |
| 3300                | $10^{-9}$                            | 10                   | 123,635                               |
| 3350                | $10^{-8}$                            | 10                   | 127,182                               |

Fonte: Produzido pelo autor

Isto posto, considerando que o *hardware* utilizado era doméstico, pode-se concluir que o desempenho do método de Newton foi uma ordem de grandeza superior ao Método do Gradiente.

Portanto, *hardwares* doméstico similares ao descrito neste trabalho, podem ser utilizados para resolver problemas quadráticos irrestritos com até 3350 variáveis (e com até 8 casas decimais de precisão) sem demandar um custo temporal muito alto.

## Conclusões

Este trabalho é fruto da monografia realizada pelo autor 1 no ano de 2021. E como exposto nas linhas anteriores utilizou o Métodos do Gradiente e o de Newton para resolver problemas de programação quadrática irrestrita, via implementação computacional por meio da linguagem Python. Os testes realizados permitiram exemplificar e auxiliar o entendimento de aspectos teóricos e computacionais relacionados a classe de problemas de programação quadrática.

## Referências

- FRIEDLANDER, A. *Elementos de programação não linear*. Campinas: Pontes, 1994.
- GOLDBARG, M. C.; LUNA, H. P. L. *Otimização Combinatória e Programação Linear: modelos*

e algoritmos. 2ed. ed. Rio de Janeiro: Elsevier, 2005.

HILLIER, F. S.; LIEBERMAN, G. J. *Introdução a Pesquisa Operacional*. Tradução Ariovaldo Griesi. Revisão técnica Pierre J. Ehrlich. 9. ed. Porto Alegre: AMGH, 2013.

LUENBERGER, D. G.; YE, Y. *Linear and Nonlinear Programming*. New York: Springer International Publishing Switzerland, 2016.

MACHADO, Alex Augusto Nunes. *Otimização Irrestrita: conceitos introdutórios e exemplos numéricos com python*. 2021. 74 f. TCC (Graduação) - Curso de Matemática, Centro de Ciências Exatas e Tecnológicas, Universidade Estadual do Oeste do Paraná, Cascavel, 2021.

MARTÍNEZ, J. M.; SANTOS, S. A. Métodos computacionais de otimização. In: COLÓQUIO BRASILEIRO DE MATEMÁTICA, 20., 1995, Rio de Janeiro. *Anais...* Rio de Janeiro: IMPA, 1995. Disponível em: [https://impa.br/wp-content/uploads/2017/04/20\\_CBM\\_95\\_03.pdf](https://impa.br/wp-content/uploads/2017/04/20_CBM_95_03.pdf). Acesso em: 25 jan. 2021.

MARTINS, M. H. S.; PEREIRA, R. *Cálculo III e IV*. Florianópolis: UFSC/EAD/CED/CFM, 2010.

OTIMIZAR. In: DICIO: Dicionário Online de Português. 7Graus, 2021. Disponível em: <https://www.dicio.com.br/>. Acesso em: 21 jan. 2021.

RIBEIRO, A. A.; KARAS, E. W. *Otimização contínua: aspectos teóricos e computacionais*. São Paulo: Cengage Learning, 2013.

## Geometria Analítica em Motor de Jogos

Marco Antonio Damo  
Universidade Estadual do Oeste do Paraná  
marco.damo@unioeste.br

Flavio Roberto Dias Silva  
Universidade Estadual do Oeste do Paraná  
frdsilva@yahoo.com.br

**Resumo:** O presente artigo tem por finalidade apresentar as aplicações feitas do trabalho de iniciação científica voluntária e tem por finalidade apresentar as aplicações da geometria analítica no motor de jogos Unity 3D com programação orientada a objetos.

**Palavras-chave:** geometria analítica; motor de jogos; programação orientada a objetos.

### 1 Introdução

O trabalho proposto surge do estudo e implementação acerca da geometria analítica dentro da programação orientada a objetos. O trabalho tem como eixo o seguinte questionamento: Como pode ser utilizada essa área da matemática para programar um jogo?

O objetivo geral é compreender sobre o desenvolvimento da movimentação em jogos, em particular, na programação. Diante desse objetivo geral, o objetivo específico é fazer um estudo de caso, por meio da programação orientada a objetos, como se comportam os vetores dentro de um jogo.

Durante o trabalho foram feitos estudos relacionados a geometria analítica utilizando as bibliografias (STEINBRUCH, 1987), que auxiliaram no aprendizado de uma forma diferente de programar, unindo fortes áreas como a programação e a matemática, dando auxílio para o melhor conhecimento dos resultados obtidos nas implementações.

A natureza desse trabalho é de cunho qualitativo e explicativo, utilizando do estudo de caso para realizar seu encadeamento rumo ao seu epílogo.

### 2 Unity

Atualmente, existem inúmeros motores de jogos em 3D que auxiliam no desenvolvimento de games, e entre eles, há um que será o personagem principal no que nos diz respeito a implementação dos estudos feitos por essa iniciação científica, a Unity.

A Unity é um motor de jogos muito popular atualmente, é gratuito e por isso atrai muitas pessoas que são incentivadas em ingressar nos seus estudos. Ela tem suporte para jogos 2D e

3D, além da compilação para vários sistemas operacionais como Windows, Linux, Android, entre outros. Os nossos estudos direcionados para Unity foram feitos com base na documentação online que pode ser consultada em (UNITY TECNOLOGIES, 2021(A)), (UNITY TECNOLOGIES, 2021(B)), (UNITY TECNOLOGIES, 2021(C)), (UNITY TECNOLOGIES, 2021(D)).

Nesta pesquisa, foi utilizada como linguagem de programação o C# na Unity, portanto, as definições serão feitas no contexto dessa linguagem, como pode ser consultado em (STELLMAN, 2008). A quantidade de funcionalidades é enorme, pode-se passar páginas apenas da explicação de uma delas. Isto posto, serão abordadas as principais e mais utilizadas para essa pesquisa.

O C# é uma linguagem de programação orientada a objetos e é a linguagem principal do .NET Framework, que é a principal plataforma para desenvolvimento de programas da Microsoft.

## 2.1 Arquitetura da Programação

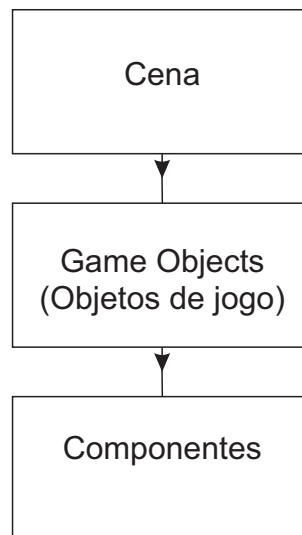


Figura 1: Arquitetura da Programação  
Fonte: Produção Própria com auxílio do VSCode

A arquitetura da programação da Unity é organizada como na imagem, o jogo é programado em forma de cenas, dentro das cenas temos gameObjects e, anexados aos gameObjects, temos componentes. Os scripts que escrevemos para programar o jogo entram em cena anexados a gameObjects e na forma de componentes.

### Cena

A cena é uma representação computacional do mundo 3D simulada pelo motor de jogos. Na interface Unity temos uma representação visual da cena atualmente carregada que nos permite fazer diversas edições nos elementos do jogo.

## GameObject

O gameObject é a classe base de todas as entidades do jogo, ele pode assumir formas, responsabilidades, diferentes hierarquias, entre outros. Os gameObjects são terrenos, personagens, armadilhas, luzes e cameras, objetos vazios, entre outros.

## Componentes

Os componentes são responsáveis pelas ações, limitações e características que serão atribuídas a um gameObject. Os scripts que escrevemos são adicionados em gameObjects como componentes.

### 2.2 Script Vazio

Um script pode ser criado na Unity através de comandos simples e quando ele é criado, segue um padrão pré-definido pelo motor de jogos:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class exemplo : MonoBehaviour
6  {
7      // Start is called before the first frame update
8      void Start()
9      {
10
11      }
12
13      // Update is called once per frame
14      void Update()
15      {
16
17      }
18  }
```

Figura 2: Exemplo de Script Vazio

Fonte: Produção Própria com auxílio do VSCode

Abaixo daremos uma breve descrição dos elementos constantes nesse script base da Unity.

## Bibliotecas

System.Collections é uma biblioteca de coleções nativa do C#. Já a UnityEngine é a biblioteca com os elementos internos do motor de jogos, ela nos dá acesso a maioria das estruturas e classes que podem ser utilizadas no desenvolvimento de um jogo.

## **MonoBehaviour, Start e Update**

As classes são um conjunto de métodos e propriedades que estabelece um comportamento no desenvolvimento de um software.

Primeiramente, MonoBehaviour é uma classe que faz parte da UnityEngine e é responsável por tudo o que ocorre em tempo de jogo, como a chamada das funções Start e Update, detector de colisões, inserir e destruir objetos de jogo na cena, entre outros.

No nosso script base, podemos observar dois métodos, o Start e o Update. Esses são os dois métodos mais utilizados na programação de um jogo na Unity, tendo o primeiro o papel de inicializar um objeto e o segundo de atualizar esse objeto a cada quadro de jogo.

Start é chamada no momento em que o objeto no qual foi anexado esse script entrar em cena e Update é chamada uma vez por quadro, que se baseia na quantidade de fps (quadros por segundo); por exemplo, 30 fps fará com que ela seja chamada 30 vezes em um segundo.

## **2.3 Estruturas e Classes**

No que segue, mencionaremos algumas classes e estruturas da Unity Engine que se farão presentes na nossa discussão principal sobre as operações básicas da geometria analítica dentro do motor de jogos.

### **Estrutura Vector3**

É uma estrutura que guarda três valores de ponto flutuante que representam direções ou pontos no espaço 3D e é utilizado principalmente para manipular posições e direções no plano tridimensional, tanto quanto operações vetoriais comuns.

### **Classe Input**

É a classe responsável pelas entradas de informações como a leitura do teclado, do mouse, joystick, entre outros. É uma das classes essenciais para o desenvolvimento de um jogo.

### **Componentes Transform, Rigidbody e Collider**

O Transform é utilizado para armazenar e manipular a posição, rotação e escala do gameObject. Todo script que herda da classe MonoBehaviour tem uma propriedade transform que é essencial para fazer a movimentação do objeto no espaço 3D da Unity.

O Rigidbody é a classe responsável por simular a física no ambiente 3D da Unity. Entre as principais características dessa simulação estão a resposta a gravidade, resposta a colisões, simulação de forças de impulso, explosões e etc.

Já o Collider é um componente responsável por construir a caixa de colisão de todos os objetos, e é essencial para a simulação das físicas em jogo com o Rigidbody.

```
1 public Vector3 nv = new Vector3(1,2,0);
2
3 void Start()
4 {
5     transform.position = nv;
6 }
```

Figura 3: Exemplo da Utilização de alguns conceitos

Fonte: Produção Própria com auxílio do VSCode

Nesse pequeno código, está sendo declarada uma variável com o nome “nv” do tipo Vector3, que está recebendo 3 coordenadas que representarão uma posição no espaço. Após isso, observamos que ao entrar na cena, a função Start é chamada, assim, o transform no seu campo posição recebe a posição armazenada pela variável “nv” e isso desloca o objeto para a posição (1, 2, 0) na simulação 3D do jogo.

```
1 public class Movimentação : MonoBehaviour
2 {
3     public Rigidbody jogador;
4
5     // Start is called before the first frame update
6     void Start()
7     {
8         jogador = GetComponent<Rigidbody>();
9     }
10
11     // Update is called once per frame
12     void Update()
13     {
14         float cordx = Input.GetAxis("Horizontal");
15         float cordz = Input.GetAxis("Vertical");
16         Vector3 movement = new Vector3(cordx, 0, cordz);
17         movement *= vel * Time.deltaTime;
18     }
19 }
20
```

Figura 4: Exemplo de GetComponent, GetAxis e Time.deltaTime

Fonte: Produção Própria com auxílio do VSCode

## 2.4 GetComponent

No código da figura 4, podemos observar que a variável chamada jogador está recebendo o retorno da função GetComponent. A função GetComponent pega o componente passado como parâmetro entre os brackets( ) e está anexado ao mesmo gameObject no qual este script está anexado. No caso, a função retorna uma referência para o componente Rigidbody deste gameObject e se não houver Rigidbody anexado ao gameObject a função retornará nulo.

## 2.5.GetAxis e Time.deltaTime

O GetAxis é uma função da classe input (Input.GetAxis) responsável por retornar o valor do eixo virtual pré-definido pela Unity, esse será um valor de ponto flutuantes que estará entre -1 e 1. As teclas responsáveis por esse input podem ser modificados por um painel chamado “Input Manager” e dependendo do tipo de controle de entrada o programa reconhecerá 1 como sendo empurrado totalmente para a direita e -1 totalmente para a esquerda.

É importante ter em mente que as bibliotecas da Unity já contêm certos parâmetros pré-definidos, como “Vertical” e “Horizontal”, que são reconhecidos pelas teclas “w”, “a”, “s” e “d” ou as setas encontradas, na maioria dos casos, no lado inferior direito inferior do teclado ou por outros dispositivos externos.

Já o Time.deltaTime é responsável por ler o intervalo em segundos do último quadro até o atual. No nosso caso ele é utilizado para aplicar a velocidade por segundo em relação ao update que atualiza uma vez por quadro.

## 3 Script

A área de programação vem crescendo cada vez mais nas últimas décadas, e os ocupantes dos maiores pódios são estusiasistas que desenvolvem os melhores Scripts, mas afinal, o que realmente é “Script”?

De maneira direta, scripts são ?roteiros? seguidos por sistemas computacionais e trazem informações que são processadas e transformadas em ações efetuadas por um programa principal.(PEREIRA, 2012).

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class Movimentação : MonoBehaviour
6  {
7      public float vel = 10;
8      public Rigidbody jogador;
9
10     // Start is called before the first frame update
11     void Start()
12     {
13         jogador = GetComponent<Rigidbody>();
14     }
15
16     // Update is called once per frame
17     void Update()
18     {
19         float cordx = Input.GetAxis("Horizontal");
20         float cordz = Input.GetAxis("Vertical");
21         Vector3 movement = new Vector3(cordx, 0, cordz);
22         movement *= vel * Time.deltaTime;
23         jogador.velocity = movement;
24     }
25 }
```

Figura 5: Script de Movimentação

Fonte: Produção Própria com auxílio do VSCode

Podemos observar que as bibliotecas utilizadas são as padrões para o desenvolvimento de um jogo em Unity com a utilização de C#. A partir disso a classe pública que herda `MonoBehaviour` recebeu o nome de `Movimentação`, pois essa será a responsabilidade desse script. As variáveis definidas fora do `Start` e `Update` foram o valor de velocidade, que foi declarado por uma variável pública do tipo ponto flutuante recebendo o valor dez e o jogador que é uma variável do tipo `Rigidbody` com intuito de operar respeitando as leis da física no espaço tridimensional.

Ao entrar na cena (`Start`), a variável jogador é definido como o componente `Rigidbody`. Depois disso, o `Update` é responsável por estar definindo as interações em tempo de jogo, é dentro dele onde são realizadas as operações responsáveis pelas características da movimentação.

Oberva-se no início do método `Update` a declaração de duas novas variáveis, a `cordx` e `cordz`. É atribuído a elas o `Input.GetAxis`. Portanto, a partir de agora a variável “`cordx`” e a variável “`cordz`” recebe um valor entre -1 e 1 que corresponde às direções horizontal ou vertical pré-definidas pela Unity.

Na linha seguinte do código, é declarada uma variável do tipo `Vector3` que recebe o nome de “`movement`” e é atribuída um vetor de 3 posições com os valores de  $x = \text{cordz}$ ,  $y = 0$  e  $z = \text{cordz}$ . O  $y$  recebe 0 pois este eixo é vertical no mundo tridimensional e o objetivo é fazer uma movimentação sem deslocamento vertical, portanto as coordenadas se verificam nas dimensões  $x$  e  $z$ .

Em seguida, o vetor responsável pela movimentação necessita de uma velocidade mais realista, por isso foi atribuído a velocidade que foi definida como “`vel`” de valor 10 multiplicada

pelo `Time.deltaTime` para que a velocidade `vel` seja aplicada em 10 unidades por segundo no jogo com auxílio da função `Update`. Com essa interação, enquanto o input é pressionado a velocidade irá aumentar progressivamente. Por fim temos o `velocity`, que é o vetor velocidade do `rigidbody`, e ele representa a taxa de mudança da posição do jogador.

## 4 A influência da Geometria Analítica

Finalmente, nessa seção, mostraremos como as operações básicas da geometria analítica são aplicadas na programação de jogos 3D.

Essa grande área da matemática é responsável por estudar o comportamento dos vetores que são utilizados como ferramenta para explicar diversos fenômenos naturais e artificiais, como por exemplo, a aplicação de forças em um corpo e como agem causando variação no movimento.

Os elementos de um jogo são movimentados através de operações vetoriais, alguns desses elementos se propõem a simular a física do mundo real e são movimentados a partir da simulação da aplicação de uma força. Quando o input é pressionado enquanto o script está agindo, o motor de jogos receberá essa informação e movimentará o `gameObject` no qual está anexado ao script. Essa interação muda a posição do personagem a qual é armazenada como uma variável do tipo `Vector3` dentro do componente `transform`. Essa operação é realizada repetidamente a cada quadro enquanto o input está ativo. Essa interação pode ser explicada pela soma de ponto com vetor.

**Definição 1** (Soma de Ponto com Vetor). Quando se tem um ponto  $A$  e um vetor  $\vec{v}$  existe um único ponto  $X$  tal que  $\vec{AX} = \vec{v}$ . Chamaremos o ponto  $X$  de soma do ponto  $A$  com o vetor  $\vec{v}$  (STEINBRUCH; WINTERLE, 1987).

A informação de posição guardada no `transform`, como nos nossos exemplos, é atualizada com a ação dos inputs dentro da função `update`. Em cada interação temos uma modificação da posição do `gameObject` no espaço tridimensional, assim realizando a movimentação desejada.

```
21      Vector3 movement = new Vector3(cordx, 0, cordz);  
22      movement *= vel * Time.deltaTime;  
23      jogador.velocity = movement;
```

Figura 6: Script de Movimentação: vetores  
Fonte: Produção Própria com auxílio do VSCode

**Definição 2** (Produto de vetor por escalar). Dado um vetor  $\vec{v}$  e um número real  $k$ , chama-se produto do número real  $k$  pelo vetor  $\vec{v}$  o vetor  $\vec{p} = k\vec{v}$  definido por:

- Se  $\vec{v} = \vec{0}$  ou  $k = 0$  definimos  $k\vec{v} = \vec{0}$
- Se  $\vec{v} \neq \vec{0}$  e o número real  $k \neq 0$ , o vetor é caracterizado por:

- $k\vec{v}$  tem a mesma direção de  $\vec{v}$ ;
- $k\vec{v}$  tem o mesmo sentido de  $\vec{v}$  se  $k > 0$  e sentido diferente se  $k < 0$ ;
- $|\vec{p}| = |k\vec{v}| = |k||\vec{v}|$ . (BOULOS, 1987)

No Script da figura 6 é efetuado um produto de vetor por escalar, a variável “vel” multiplicada por “Time.deltaTime” representa um número real, em seguida o vetor “movement” recebe o valor que estava guardado nele multiplicado por esse número real, aqui temos a multiplicação de vetor por escalar. Isso se repete até o momento em que o input deixa de ser pressionado e a variável movement recebe o vetor nulo  $\vec{0}$ , assim o transform guarda as novas coordenadas recebidas e o velocity armazena a taxa de mudança de posição em vetores.

Existem outras formas de reproduzir o mesmo comportamento de movimentação com alguma variação no script. Na imagem abaixo, é exemplificado uma alteração do tipo, nela podemos observar uma aplicação da soma de vetores para a construção do vetor que indica a direção do movimento.

```
11 // Update is called once per frame
12 void Update()
13 {
14     float cordx = Input.GetAxis("Horizontal");
15     float cordz = Input.GetAxis("Vertical");
16     Vector3 movement = new Vector3(cordx, 0, cordz);
17     movement = cordz*Vector3.forward+cordx*Vector3.right;
18 }
```

Figura 7: Movimentação com Soma de Vetores  
Fonte: Produção Própria com auxílio do VSCode

**Definição 3** (Adição de Vetores). Dados  $\vec{u}$  e  $\vec{v}$  em  $V^3$  definiremos a soma de vetores e indicaremos por  $\vec{u} + \vec{v}$  o vetor obtido com o seguinte processo: Consideramos um representante qualquer (A, B) do vetor  $\vec{u}$  e o representante do vetor  $\vec{v}$  que tem origem B. Seja C a extremidade deste último. Fica assim determinado o segmento orientado (A, C). Por definição, o vetor  $\vec{AC}$ , cujo representante é o segmento orientado (A, C), é o vetor soma de  $\vec{u}$  com  $\vec{v}$  (STEINBRUCH; WINTERLE, 1987).

No script da figura 7 a soma de vetores ocorre na linha 17. Nessa linha o Vector3.forward representa a direção (0, 0, 1) e Vector3.right a direção (1, 0, 0) na simulação 3D da Unity. Como mencionado anteriormente, cordz e cordx são valores preenchidos pelos inputs, com isso, cordz\*Vector3.forward e cordx\*Vector3.right, são vetores resultados do produto de vetor por escalar, e portanto, o sinal de soma na linha do script representa a soma de dois vetores conforme a definição da geometria analítica.

## Conclusões

Nesse artigo fizemos o estudo de caso de como as operações básicas da geometria analítica são efetuadas dentro do motor de jogos Unity e podemos concluir que essas operações são feitas de forma intuitiva na linguagem de programação e simulam adequadamente a representação visual dos efeitos físicos dessas operações.

## Agradecimentos

Gostaria de agradecer ao professor Flavio Roberto Dias Silva por estar me auxiliando para o melhor entendimento dos estudos, de forma profissionalmente didática.

## Referências

BOULOS, Paulo; CAMARGO, Ivan de. **Geometria Analítica: um tratamento vetorial**. São Paulo: Makron Books, 1987. 387 p.

GREENE, Jennifer; STELLMAN, Andrew. **Use a Cabeça C#**. 2. ed. Rio de Janeiro: Alta Books, 2008. 840 p.

PEREIRA, André Luiz. **O que é script?**. 2012. Elaborada por tecmundo. Disponível em: <https://www.tecmundo.com.br/programacao/1185-o-que-e-script-.htm>. Acesso em: 27 ago. 2021

STEINBRUCH, Alfredo; WINTERLE, Paulo. **Geometria Analítica**. 2. ed. São Paulo: Makron Books, 1987. 294 p.

UNITY TECHNOLOGIES (org.)(A). **Unity 3D User Manual**. 2021. Disponível em: <http://docs.unity3d.com/Manual/index.html>. Acesso em: 08 fev. 2021.

UNITY TECHNOLOGIES (org.)(B). **Unity 3D Community Forum**. Disponível em: <http://forum.unity3d.com/>. Acesso em: 08 fev. 2021.

UNITY TECHNOLOGIES (org.)(C). **Unity 3D Online Tutorials**. Disponível em: <https://unity3d.com/pt/learn/tutorials>. Acesso em: 08 fev. 2021.

UNITY TECHNOLOGIES (org.)(D). **Unity 3D Community Wiki**. Disponível em: [http://wiki.unity3d.com/index.php/Main\\_Page](http://wiki.unity3d.com/index.php/Main_Page). Acesso em: 08 fev. 2021.

## O Problema do Caixeiro Viajante

Eduardo Ghisi<sup>1</sup>

Universidade Estadual do Oeste do Paraná  
eduardoghisi66@outlook.com

Simone Aparecida Miloca

Universidade Estadual do Oeste do Paraná  
simone.miloca@unioeste.br

**Resumo:** Um dos problemas clássicos de otimização combinatória é o Problema do Caixeiro Viajante, que devido a sua natureza e aplicações é assunto presente na literatura da área. O objetivo deste trabalho é apresentar uma de suas formulações matemáticas, gerar algumas instâncias de teste e resolvê-las utilizando um dos *solvers* que tem ganhado destaque na literatura, que é o IBM-ILOG-CPLEX Studio.

**Palavras-chave:** CPLEX Studio; Otimização Combinatória; TSP.

### 1 Introdução

Um dos problemas clássicos de otimização combinatória é o Problema do Caixeiro Viajante (PCV) - *Travelling Salesman Problem (TSP)*. O problema consiste em partir de uma cidade origem e visitar uma única vez cada uma das  $(n - 1)$  cidades restantes e retornar à cidade origem. O trajeto escolhido deve ser o que minimize uma função relacionada a distância total percorrida ou ao custo do trajeto ou ao tempo da viagem.

O interesse pelo PCV é devido aos inúmeros problemas reais que são modelados como problemas do tipo caixeiro viajante e suas variantes tais como problemas de roteamento de veículos, problemas de sequenciamento de tarefas, planejamento da produção e programação de operações em máquinas de manufaturas conforme descrito em Goldberg e Luna (2005).

Além das aplicações, o problema tem sido objeto de estudo para muitos pesquisadores por pertencerem a classe de problemas de otimização combinatória. A Figura 8 exibe uma classificação de tais problemas, que se enquadra como um problema de Programação Inteira Mista (HILLIER e LIEBERMAN, 2013).

---

<sup>1</sup>Agradecemos o apoio da Fundação Araucária

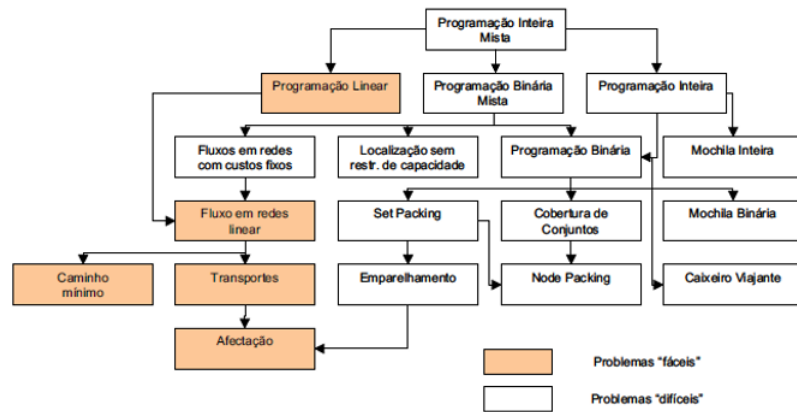


Figura 8: Problemas de otimização combinatória e complexidade.

Fonte: Nemhauser and Wolsey - Wiley, 1988.

Devido a esta e outras características, o PCV é um dos problemas de otimização combinatória mais estudados na literatura, tendo diversas abordagens de solução propostas. Segundo Hillier e Lieberman (2013), poderosos algoritmos baseados em metodologias de ramificação e avaliação progressiva conseguem encontrar a otimalidade para certos problemas com várias centenas ou até milhares de cidades. Outro enfoque presente na literatura é a utilização de métodos heurísticos orientados pelas metaheurísticas (LOPES *et al.*, 2013).

A programação inteira mista foi investigada extensivamente nas últimas décadas e bons *solvers* estão agora disponíveis para várias classes de programações inteiras (KRONKVIST *et al.*, 2019). Neste trabalho propõe-se tomar uma das formulações do PCV (caso simétrico), gerar algumas instâncias e resolvê-las utilizando-se o software IBM-ILOG-CPLEX Studio.

## 2 Problema do Caixeiro Viajante

Considere um conjunto de  $n$  cidades,  $N = \{1, 2, 3, \dots, n\}$ , e uma matriz de distâncias com informação de ligação de cada par de cidades de  $N$ . O problema do Caixeiro Viajante consiste em partir de uma cidade origem e visitar uma única vez cada uma das  $(n - 1)$  cidades restantes e retornar à cidade origem minimizando uma função relacionada a distância total percorrida ou ao custo do trajeto ou ao tempo da viagem. Todas as cidades são ligadas por estradas e a rota pode iniciar em qualquer uma das cidades. A rota é representada utilizando-se um grafo completo  $G(N, A)$  onde  $N$  é o número de nós, representando as cidades e  $A$ , o número de arestas, indicando os caminhos entre as cidades. O problema é denominado simétrico quando supõe-se que a distância é a mesma em ambas as direções.

Dentre as formulações existentes, o modelo matemático que envolve a eliminação de sub-rotas e visa minimizar as distâncias é dado como segue:

$$\text{minimizar} \quad f(x) = \sum_{j=1}^n \sum_{i=1}^n c_{ij} x_{ij} \quad (1)$$

$$\text{sujeito a} \quad \sum_{i=1}^n x_{ij} = 1, \quad j = 1, \dots, n \quad (2)$$

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n \quad (3)$$

$$u_i - u_j + n \cdot x_{ij} \leq n - 1, \quad i = 2, \dots, n, \quad j = 2, \dots, n, \quad i \neq j \quad (4)$$

$$x_{ij} \in \{0, 1\} \quad (5)$$

$$u_i \in \mathbb{R}^+. \quad (6)$$

onde,

$$x_{ij} = \begin{cases} 1 & \text{se o caixeiro vai diretamente da cidade } i \text{ para a cidade } j \\ 0 & \text{caso contrário} \end{cases}$$

Esta formulação pode ser vista em Lopes *et al.* (2013). Cabe lembrar que nesse contexto a preocupação é minimizar a distância sem observar o tempo e o custo. A função objetivo (1) minimiza a soma das distâncias ( $c_{ij}$ ) entre as cidades  $i$  e  $j$ . As restrições (2) e (3) impõem que para cada cidade  $j$ , o caixeiro só poderá chegar de uma única cidade  $i$  e de cada cidade  $i$ , o caixeiro só poderá ir para uma única cidade  $j$ . As restrições (4) referem-se à eliminação de sub-rotas, onde as variáveis  $u_i$  e  $u_j$  são auxiliares e não tem um significado físico. As restrições (5) e (6) indicam o tipo de variáveis.

Observe que é possível calcular o número de caminhos pois o caixeiro parte de uma das  $n$  cidades, dali ele segue para qualquer uma das  $(n - 1)$  cidades e assim por diante totalizando  $n(n - 1)(n - 2) \cdots 2 \cdot 1 = n!$  rotas possíveis. Esse tipo de problema possui domínio finito e em geral, pode-se listar os seus elementos e testar se um dado elemento pertence a esse domínio. No entanto, na prática, testar todos os elementos na busca pelo melhor pode ser inviável. Neste caso, diz-se que o problema é intratável (difícil), ou seja, o espaço solução pode ser tão amplo, que para percorrê-lo precisaríamos de um tempo tão longo que pode-se considerar que o problema não tem solução através de métodos de enumeração exaustiva (ARENALES *et al.*, 2015).

O PCV é formulado como um problema de programação inteira ou como inteira mista (PIM). Possui complexidade exponencial que faz com que se enquadre na classe de problemas “difíceis”. O método mais utilizado para resolver um problema desta natureza é o *Branch-and-Bound* (B&B) e suas variantes, detalhado em Hillier e Lieberman (2013).

Um dos *solvers* apontado na literatura que produz bons resultados é o CPLEX que contém solucionador para problemas de programação inteira mista. Seu algoritmo incorpora métodos de Pontos Interiores e *Branch-and-Bound* e manipula com rapidez problemas de grande porte. Na seção seguinte apresentamos alguns testes numéricos realizados com este *solver* para resolver algumas instâncias do PCV.

### 3 Testes Numéricos com IBM-ILOG-CPLEX Studio

O nome CPLEX refere-se a linguagem de programação C e ao método Simplex, único método de solução disponível na época em que foi desenvolvido. Segundo o manual IBM (2021),

a versão atual, além de solucionar problemas de programação linear, também inclui uma classe particular de programação não-linear, a programação quadrática (QP). Também possui solucionador para programação inteira mista, tanto do tipo PL ou QP. A ILOG oferece licenças gratuitas para fins acadêmicos. Deve-se acessar o site e fazer um cadastro no portal da IBM<sup>2</sup>.

Testes numéricos foram realizados no sentido de observar o que ocorre quando se aumenta o número de cidades. O PCV foi implementado utilizando o OPL como uma linguagem de modelagem e o IBM ILOG Script for OPL como uma linguagem de script. Foram geradas cinco instâncias e usadas como dados de entrada para o modelo genérico. Tais dados são o número de cidades e a respectiva matriz de distância que será fornecida no CPLEX utilizando o ambiente .dat, sendo o modelo geral apresentado na Figura 9.

|                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                           |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>// Cidades int    n      = ...; range  cidades = 1..n;  float dist[cidades][cidades] = ...;  // Variáveis de decisão dvar boolean x[cidades][cidades]; dvar float+ f[cidades];  //MODELO  // Função Objetivo minimize sum (i in cidades , j in cidades : i!=j)     dist[i][j]*x[i][j];</pre> | <p><b>Continuação do código do modelo</b></p> <pre>//Restrições subject to {     forall (i in cidades)         sum (j in cidades: i!=j) x[i][j] == 1;      forall (j in cidades)         sum (i in cidades: i!=j) x[i][j] == 1;      //Eliminacao de sub-rotas     forall (i in cidades , j in cidades: i &gt; 1 &amp;&amp; j &gt; 1 &amp;&amp; i!=j)         f[i] - f[j] + n*x[i][j] &lt;= n-1; };</pre> |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Figura 9: Implementação do modelo.

Fonte: O autor.

Primeiro foram obtidas randomicamente as coordenadas de 05 cidades, depois para 10, 20, 30 e 31, sendo que a versão de estudante do software CPLEX, por nós utilizada, permitiu resolver até o número máximo de 31 cidades.

A Tabela 1 apresenta os resultados. Observe que conforme se aumenta o número de cidades, o número de restrições e iterações cresce bastante. Note que o tempo também cresce consideravelmente, veja o caso de 30 para 31 cidades.

Tabela 1: Resultados fornecidos pelo CPLEX para o problema do caixeiro viajante.

| n° de cidades | n° de variáveis | n° de restrições | n° de iterações | função objetivo | tempo |
|---------------|-----------------|------------------|-----------------|-----------------|-------|
| 5             | 24              | 22               | 20              | 165,693         | 00,05 |
| 10            | 99              | 92               | 46              | 266,394         | 00,16 |
| 20            | 399             | 382              | 132             | 372,816         | 00,25 |
| 30            | 899             | 872              | 23945           | 423,534         | 04,14 |
| 31            | 960             | 932              | 27384           | 441,984         | 09,70 |

Nota: A unidade de medida do tempo é dada em segundos.

A Figura 10 mostra as rotas ótimas para o caso de  $n = 20$  e  $n = 31$  cidades.

<sup>2</sup>Link para acesso ao site do portal IBM: <https://www.ibm.com/br-pt>

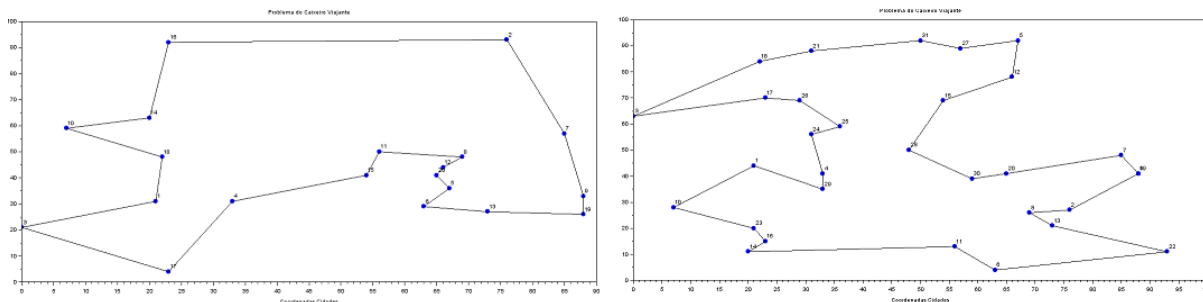


Figura 10: Rotas ótimas do PCV para 20 cidades (figura da esquerda) e 31 cidades (figura da direita).

Fonte: O autor.

## Conclusões

Através das implementações foi possível observar como pode ser difícil encontrar soluções para problemas desta natureza. Lembre-se que, no exemplo apresentado nesse estudo, o tempo e o número de iterações aumentou consideravelmente ao comparar 30 com 31 cidades. Quanto maior o número de variáveis inteiras maior será o tempo para encontrar a solução ótima. Como trabalho futuro pretende-se realizar testes com um número maior de cidades. Este trabalho faz parte de estudos realizados durante o projeto de iniciação científica desenvolvido na Universidade Estadual do Oeste do Paraná.

## Referências

- ARENALES, Marcos; ARMENTANO, Vinícius; MORABITTO, Reinaldo; YANASSE, Horacio. Pesquisa Operacional. 2. ed. Rio de Janeiro: Campus, 2015.
- GOLDBARG, Marco Cesar; LUNA, Henrique Pacca L.. Otimização Combinatória e Programação Linear. 2. ed. Rio de Janeiro: Elsevier, 2005.
- HILLIER, Frederick S.; LIEBERMAN, Gerald J.. Introduction to Operations Research. 8. ed. New York: McGraw-Hill, 2013. 1005 p.
- IBM. Manual do usuário. 2021. Disponível em: <https://www.ibm.com/docs/pt-br/icos/12.9.0?topic=cplex-users-manual>. Acesso em: 15 fev. 2021.
- KRONKVIST, Jan; BERNAL, David E.; LUNDEL, Andreas; GROSMANN, Ignacio E.. A review and comparison of solvers for convex MINLP. Optimization and Engineering, Springer, 2019.
- LOPES, Heitor Silverio; RODRIGUES, Luiz Carlos de Abreu; STEINER, Maria Terezinha Arns (ed.). Meta-Heurísticas em Pesquisa Operacional. Curitiba: Omnipax, 2013.
- NEMHAUSER, George; WOLSEY, Laurence. Integer and Combinatorial Optimization. New York: John Wiley, 1988. 784 p.

## O Uso de Novas Tecnologias no Ensino Remoto nas Aulas de Matemática: Relato de Experiência.

Fernanda Carla de Oliveira  
Universidade Estadual do Oeste do Paraná  
fernanda.oliveira12@unioeste.br

Nadya Beatriz Antunes Barroso  
Universidade Estadual do Oeste do Paraná  
nadya.barroso@unioeste.br

Pamela Goncalves  
Universidade Estadual do Oeste do Paraná  
pamela.goncalves@unioeste.br

**Resumo:** O artigo relata a experiência de estágio supervisionado no ensino remoto, durante o período emergencial da pandemia do Covid – 19, que não afetou somente a educação, como o mundo inteiro. No decurso da elaboração das atividades, tínhamos como intenção desvendar as dificuldades e pontos positivos para esse método implantado, que durante essa pandemia supre a necessidade da educação no momento de distanciamento social. A busca de novas tecnologias e plataformas para o desenvolvimento de atividades durante às aulas, foi o ponto chave para criação do plano de aula apresentado, cujo objetivo era revisar os conteúdos trabalhados pela professora regente. O relato em si é referente a uma aula de matemática para alunos do segundo ano do ensino técnico em edificações do Instituto Federal do Paraná – IFPR do campus de Cascavel, abordando os conteúdos de círculo e razões trigonométricas. Os resultados obtidos destacam fatores relevantes que nos permitiram identificar que o uso da tecnologia e plataformas interativas pode proporcionar maior motivação, colaboração e participação dos alunos nas aulas remotas.

**Palavras-chave:** Ensino remoto; Relato de experiência; Novas Tecnologias.

### 1 Introdução

A Organização Mundial de Saúde (OMS) caracterizou em 11 de março de 2020 como pandemia, o quadro da disseminação comunitária da Covid-19. Os primeiros casos de pessoas contaminadas pelo vírus no Brasil foram em março de 2020 se estendendo até os dias de hoje. Algumas medidas foram adotadas a fim de combater o vírus, entre elas isolamento social, tratamento dos casos identificados, testes massivos e impossibilidade de atividades presenciais.

Durante esse momento crítico que estamos passando, a educação e o processo de ensino aprendizagem sofreram inúmeras mudanças. O distanciamento social, implicou no fechamento de espaços públicos, incluindo as instituições escolares. Não sendo possível antever todos os impactos negativos causados pela pandemia de Covid-19 na educação brasileira, além de supor que podem ser graves e duradouros, segundo relatório do Banco Mundial.

A partir da portaria nº 544, de 16 de junho de 2020 o Ministério de Educação do Brasil, dispõe a substituição das aulas presenciais por aulas em meios digitais, enquanto durar a situação de pandemia do novo coronavírus - Covid19 (BRASIL, 2020). Com isso, tivemos que adaptar a forma com que trabalhamos, estudamos e convivemos em sociedade. A implementação das tecnologias digitais para o desenvolvimento do conhecimento em sala de aula, se torna um grande aliado no espaço remoto e consequentemente dos professores conforme Chaves (2017, p. 3),

o que é particularmente fascinante nas novas tecnologias disponíveis hoje, em especial na Internet, e, dentro dela, na web, não é que, com sua ajuda, seja possível ensinar remotamente ou à distância, mas, sim, que elas nos ajudam a criar ambientes ricos em possibilidades de aprendizagem nos quais as pessoas interessadas e motivadas podem aprender quase qualquer coisa sem, necessariamente, se envolver num processo formal e deliberado de ensino.

Muitos desafios surgiram com a necessidade do ensino remoto, além das dificuldades com as tecnologias, falta de capacitação dos professores, é necessário garantir a interação com os alunos, para assegurar sua aprendizagem. A proposta de educação ofertada por meios tecnológicos sempre trouxe alguns obstáculos, principalmente pela falta de preparo/capacitação dos professores no manuseio de suportes tecnológicos. Entretanto, necessitamos da compreensão docente de que a tecnologia é o caminho fundamental para que essa transformação se efetive (ROSA, 2020).

O objetivo do artigo é relatar a experiência de estágio supervisionado, referente à aula ministrada para os alunos do segundo ano do curso técnico em edificações IFPR (Instituto Federal do Paraná), no ano letivo de 2021. Com o objetivo de usar as tecnologias como meio de produção de conhecimento matemático no ensino-aprendizagem dos alunos, despertando o interesse e a adaptação a essa nova realidade.

## 2 Desenvolvimento

No ensino presencial, sentimos falta da inclusão de novas tecnologias que ajudam no desenvolvimento do conteúdo para o aluno, já quando o cenário muda para o ensino remoto, temos o dever de buscar novas soluções para que o aprendizado não seja defasado. Criar uma ponte entre o conhecimento e as novas tecnologias não se torna uma tarefa fácil, e como na educação tudo se tornou novo, conforme discutem Engelbrecht, Llinares e Borba (2020, p. 836):

Muitos professores não têm a mesma experiência de ensino online como têm de ensino presencial e, de repente, há muitos “especialistas” dando conselhos sobre como uma abordagem online deve ser empregada. Os professores encontram novos problemas e se sentem um tanto isolados e desconfortáveis no ambiente. Os professores não têm certeza sobre o nível de comprometimento dos alunos com a aprendizagem (ENGELBRECHT; LLINARES; BORBA, 2020, p. 836, tradução nossa).

Dando início a realização do planejamento das aulas, foi necessária a utilização de novas tecnologias, proporcionando aos alunos uma abordagem e conhecimento de diferentes maneiras

de se trabalhar com os conteúdos de matemática, visto que as aulas remotas trouxeram um pouco de dúvidas quanto ao conhecimento adquirido pelos discentes.

Na aula do dia 14/06/2021 trabalhamos com os alunos o assunto de Circunferência Trigonométrica, organizamos um conjunto de atividades interativas utilizando as plataformas Phet Interactive Simulations e Nearpod, objetivando realizar uma revisão referente aos conteúdos ministrados anteriormente pela professora regente, integrando os alunos a um ambiente mais dinâmico e tecnológico.

O horário disponibilizado para realização do estágio foi divergente do horário das aulas síncronas da turma, realizamos no horário de atendimento da professora regente das 16 h às 17 h. Até essa data, os alunos não eram obrigados a participar das aulas síncronas, a instituição realizava o controle de presenças por meio da entrega das atividades e avaliações.

Com intuito de incentivar a participação, encaminhamos para o e-mail dos alunos e na plataforma Google Classroom no mural da turma, um convite digital conforme figura 1 abaixo, com o link de acesso para a aula síncrona por meio da plataforma Google Meet. Porém, o total de alunos presentes foi inesperado, para uma turma com 32 alunos apenas 5 estiveram presentes. Apesar, da baixa quantidade, os alunos participaram efetivamente de todas as atividades.



Figura 1: Convite para participação aula assíncrona 14/06/2021.

Fonte: Acervo das autoras

Iniciamos a aula, utilizando a Plataforma Phet que auxilia na elaboração e simulação de conteúdos matemáticos, para explicar como funciona o Círculo Trigonométrico: quadrantes, simetria dos ângulos, conversão de graus para radianos, ângulos notáveis e variação do sinal nas razões trigonométricas.

Disponibilizamos aos alunos o link de acesso à plataforma e explicamos resumidamente todas as funcionalidades da simulação, destacando para os alunos que eles utilizariam como apoio para realizar as atividades posteriores. A utilização da plataforma Phet permite ao discente simular na prática aos alunos os conceitos e conteúdos abordados em aulas teóricas.

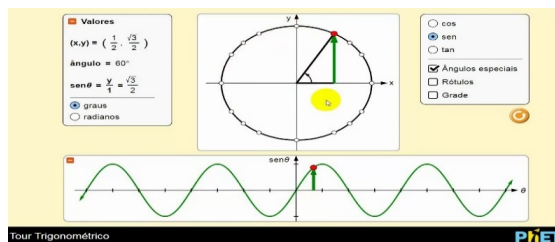


Figura 2: Tour Trigonométrico - Plataforma Phet

Fonte: Acervo das autoras

Após, disponibilizamos aos alunos o link de acesso à plataforma Nearpod, que disponibiliza jogos, atividades e apresentações educacionais. A plataforma permite a elaboração e execução de aulas interativas, o interessante dessa plataforma é que pode ser usada tanto para atividades dinâmicas, sem o caráter avaliativo, quanto para elaborações de avaliações, possibilitando ao docente acompanhar e avaliar a participação e aprendizagem dos discentes.

Para iniciar a atividade os alunos tiveram que realizar o login na página inicial do Nearpod. Intuindo evitar constrangimentos e incentivar a participação, destacamos que não havia necessidade da identificação oficial, que poderia ser escolhido um codinome aleatório.

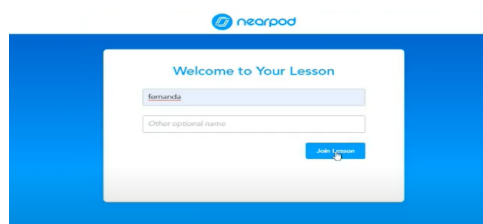


Figura 3: Página de login - Plataforma Nearpod

Fonte: Acervo das autoras

A plataforma possui várias ferramentas que possibilitam aos alunos, escrever (ferramenta texto), desenhar (ferramenta caneca) ou até mesmo adicionar uma imagem (caso tenham interesse em fazer no caderno e enviar a foto). A ideia era que eles utilizassem essas ferramentas juntamente com a plataforma Phet, para responder cada uma das questões. Inicialmente explicamos e demonstramos como utilizar as ferramentas da plataforma. Optamos por controlar as atividades, dessa forma, compartilhamos a aula com os alunos por meio de uma senha, controlamos o ritmo da apresentação, e recebemos as respostas dos estudantes em tempo real, possibilitando avaliar dúvidas, e mensurar a participação e aprendizagem.

A primeira atividade consistia em identificar e completar nos espaços vazios das figuras, para isso utilizamos a ferramenta “desenhe aqui”. Solicitamos aos discentes identificar no Ciclo Trigonométrico: os quadrantes; a variação de sinal nos quatro quadrantes para o seno, cosseno e tangente; escolher um ângulo notável no primeiro quadrante e identificar os ângulos simétricos nos demais quadrantes; e na tabela, apresentar os valores de seno, cosseno e tangente dos ângulos

notáveis de  $30^\circ$ ,  $45^\circ$ ,  $60^\circ$  e  $90^\circ$ . Conforme destacado na figura 4.

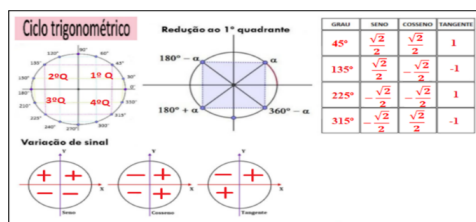


Figura 4: Atividade I – Ciclo Trigonométrico.

Fonte: Acervo das autoras

Optamos em utilizar a ferramenta de “pares correspondentes” para a atividade II, que são cartões de combinação. Os alunos deveriam relacionar os pares correspondentes na conversão de graus para radianos, como mostra a figura 5. Para isso, poderiam utilizar a plataforma Phet na confirmação das conversões, ao acertar a correspondência os pares ficam verdes, do contrário vermelhos. O professor tem acesso aos erros e acertos, possibilitando acompanhar as dificuldades e a participação dos alunos.



Figura 5: Atividade II – Pares Correspondentes.

Fonte: Acervo das autoras

A terceira atividade escolhida foi “hora de escalar”, um jogo divertido em formato de *quiz*, funciona como uma corrida. É possível escolher o cenário (optamos pelo espaço), os alunos escolhem um avatar e conforme respondem as perguntas vão subindo a montanha, chega ao topo primeiro, o aluno que responder corretamente e mais rápido dentro do tempo estabelecido. O interessante dessa atividade é que permite ao aluno acompanhar a subida de todos os participantes e ao final do jogo libera o pódio com os primeiros colocados. Conforme mostra a figura 6.

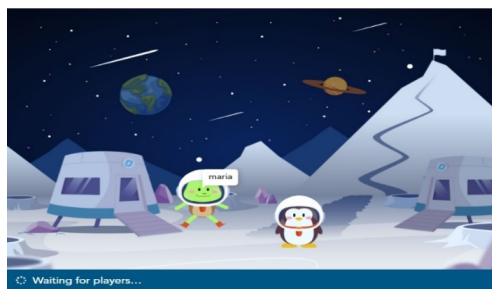


Figura 6: Atividade III - Quiz Hora da escalada.

Fonte: Acervo das autoras

Para finalizarmos a aula, utilizamos a ferramenta “painel colaborativo”, um espaço virtual para coleta de informações dos participantes, que possibilita aos alunos visualizar e interagir com os retornos dos colegas. Solicitamos aos alunos para descrever o que acharam da aula e das atividades propostas.



Figura 7: Atividade IV – Painel Colaborativo.

Fonte: Acervo das autoras

### 3 Resultados

Inicialmente os alunos tiveram dificuldades em manipular as plataformas e com a conexão da internet, o que já esperávamos por ser a primeira experiência dos alunos com esse modelo de atividade interativa. Porém, após as explicações e demonstrações, eles se adaptaram e conseguiram realizar as atividades.

De imediato não informamos aos alunos que conseguíamos visualizar o que eles estavam respondendo, na primeira atividade alguns alunos desenharam árvores e escreveram seus nomes. Brincamos com eles dizendo que os desenhos estavam bonitos, mas que preferíamos que eles participassem com as respostas. A partir daí, as atividades fluíram normalmente e foram resolvidas conforme solicitadas.

Observamos, que mesmo utilizando a plataforma Phet como auxílio para desenvolver as atividades, os alunos ainda tiveram dificuldades em compreender o que estava sendo solicitado. O conteúdo de Trigonometria é muito amplo, trabalhar esse conteúdo na modalidade remota é difícil porque envolve muitas teorias e conceitos. Porém, a possibilidade de introduzir simulações, jogos e atividades interativas colaboram para a aprendizagem do aluno.

## 4 Conclusões

A busca por novas tecnologias e plataformas se tornou algo importante para a realização do estágio, uma vez que estávamos em meio a uma pandemia, passando por um momento de novas descobertas, grandes aprendizados e grandes perdas. Iniciamos o estágio com muitas dúvidas e receios, esta foi nossa primeira experiência com o ensino remoto. Foram muitas horas de pesquisas, testes e aperfeiçoamento com as novas plataformas e tecnologias.

Os resultados obtidos destacam fatores relevantes que nos permitiram identificar que o uso da tecnologia e plataformas interativas podem proporcionar maior motivação, colaboração e participação dos alunos nas aulas remotas. Além de proporcionar a possibilidade de avaliação do desempenho e aprendizagem dos alunos por parte dos docentes.

Após a avaliação das ferramentas e metodologias utilizadas, foi possível perceber a importância para o desenvolvimento da aula, porém, é importante ressaltar que somente um tipo de metodologia ou prática pedagógica não é efetivo. Como educadores, temos que buscar inovações constantemente, buscando práticas que contribuam para a construção e conhecimento de nossos alunos.

Podemos destacar também os pontos negativos, como a desmotivação dos alunos em participar das aulas, em uma sala com 32 alunos, apenas 6 participaram da aula. Com as aulas remotas, muitos professores perceberam uma queda considerável na presença em aulas síncronas. Outro fator, que percebemos no decorrer do estágio, foi a motivação para realizar os exercícios propostos, devido a elevada falta de alunos, disponibilizamos as atividades de forma assíncrona na plataforma Google Classroom, porém como a atividade não computava nota, logo, não obtivemos retorno dos alunos. Ou seja, as atividades são realizadas somente quando valem nota.

De modo geral, a experiência apresentou resultados positivos que enfatizam a necessidade da utilização dessas e outras ferramentas pedagógicas no ensino remoto e posteriormente da inserção delas no ensino presencial.

## Referências

ENGELBRECHT, J.; LLINARES, S.; BORBA, M. C. Transformation of the mathematics classroom with the internet. **ZDM - Mathematics Education**, Karlsruhe, v. 52, n. 5, p. 825–841. Acesso em: 21 jun. 2021.

ROSA, R. T. N. da. **Das aulas presenciais às aulas remotas: as abruptas mudanças impulsionadas na docência pela ação do Coronavírus - o COVID-19!** Revista Científico Schola, Colégio Militar de Santa Maria - Santa Maria RS - Volume VI, Número 1. Acesso em: 21 jun. 2021.

CHAVES, Eduardo O C. **A Tecnologia e a Educação.** 2017. Disponível em: <https://www2.unifap.br/borges/files/2011/02/Tecnologia-na-Educacao.pdf>. Acesso em: 21 jun. 2021.

BRASIL.2020. **Ministério da Educação. Portaria nº 544, de 16 de junho de 2020** . Disponível em: <https://www.in.gov.br/en/web/dou/-/portaria-n-544-de-16-de-junho-de-2020-261924872>. Acesso em: 21 jun. 2021.

## Otimização de formas quadráticas.

Simone Aparecida Miloca

Universidade Estadual do Oeste do Paraná

smiloca@gmail.com

Paulo Domingos Conejo

Universidade Estadual do Oeste do Paraná

pconejo33@gmail.com

Alex Augusto Nunes Machado

Universidade Estadual do Oeste do Paraná

augusttoAlex@gmail.com

**Resumo:** O presente trabalho apresenta um problema de otimização restrita envolvendo formas quadráticas à luz da Álgebra Linear. Mostramos que é possível resolver o problema utilizando resultados que permitem caracterizar valores máximo e mínimo de uma forma quadrática em função de autovalores.

**Palavras-chave:** Teorema Espectral; Autovalores.

## 1 Introdução

Neste trabalho temos interesse no problema de otimização restrita à fronteira de uma bola de raio  $r$ , e que segundo Ribeiro e Karas (2013) pode ser escrito na forma,

$$\begin{array}{ll} \text{otimizar} & f(x) \\ \text{sujeito a} & x \in \Omega \subset \mathbb{R}^n, \end{array} \quad (1)$$

com  $f$  uma função  $f : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$  e  $\Omega = \{x \in \mathbb{R}^n \mid \|x\| = r\}$  o conjunto de restrições. Quando a função objetivo  $f$  é uma forma quadrática, isto é, um polinômio quadrático, o problema (1) pode ser resolvido com auxílio da Álgebra Linear, sem utilizar resultados do Cálculo, como sugere Poole (2016).

Segundo Hoffman (1979) o teorema espectral possibilita escrever um Operador Normal como combinação linear de projeções ortogonais e quando se trata de Operadores Simétricos, possibilita a classificação de cônicas, quádricas e torna a composição de funções reais mais simples. Esse processo permite ainda eliminar os termos mistos de uma forma quadrática e, desta forma concluir sobre a ocorrência de máximo e mínimo para o problema de otimização restrita (1).

Neste sentido, o presente trabalho tem por objetivo fazer uma revisão bibliográfica de elementos da Álgebra Linear como o teorema espectral e dos eixos principais, para resolver um problema de otimização restrita à fronteira de uma bola euclidiana em  $\mathbb{R}^n$ .

## 2 Formas quadráticas

Nesta seção tratamos de alguns elementos indispensáveis para o estudo de formas quadráticas e que podem ser encontrados em Hoffman (1979), Lima (1996) e Poole (2016). Realizamos uma breve apresentação de resultados da Álgebra Linear que permitem diagonalizar uma matriz simétrica, e o teorema do eixo principal, que possibilita eliminar os termos mistos em uma forma quadrática.

**Definição 1.** Considere uma matriz quadrada  $A \in \mathbb{R}^{n \times n}$ . Dizemos que um escalar  $\lambda$  é um **autovalor** de  $A$ , se existe um vetor  $v \in \mathbb{R}^n \setminus \{0\}$  tal que  $Av = \lambda v$ . Neste caso o vetor não nulo  $v$  é denominado **autovetor** de  $A$ , associado ao autovalor  $\lambda$ .

**Definição 2.** Uma matriz  $A \in \mathbb{R}^{n \times n}$  é **ortogonalmente diagonalizável** quando existem  $Q \in \mathbb{R}^{n \times n}$  ortogonal e  $D \in \mathbb{R}^{n \times n}$  diagonal tais que  $Q^T A Q = D$ .

O teorema espectral para operadores simétricos se consolida como um importante resultado da Álgebra Linear para garantir a existência de uma base ortonormal de autovetores para operadores simétricos. Detalhes em Hoffman (1979). Mas, todo operador linear em espaço de dimensão finita, pode ser representado por uma matriz. Assim, uma das consequências do teorema espectral para uma matriz simétrica, é o fato desta matriz ser ortogonalmente diagonalizável, como no teorema a seguir.

**Teorema 3.** *Considere  $A \in \mathbb{R}^{n \times n}$  simétrica. Então  $A$  é ortogonalmente diagonalizável, ou mais especificamente, existe uma matriz  $Q$  formada pelos autovetores de  $A$  e uma matriz diagonal  $D$  formada por autovalores de  $A$ , tais que  $Q^T A Q = D$ .*

*Prova.* Hoffman (1979). □

**Definição 4.** Uma forma quadrática em  $n$  variáveis é uma função  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  definida por  $f(x) = x^T A x$  com  $A \in \mathbb{R}^{n \times n}$  uma matriz simétrica.

O exemplo a seguir ilustra esta definição.

**Exemplo 1.** Considere  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$  dada por  $f(x, y) = 2x^2 - y^2 - xy$ . Podemos reescrever

$$f(x, y) = [x \ y]^T \begin{bmatrix} 2 & -\frac{1}{2} \\ -\frac{1}{2} & -1 \end{bmatrix} [x \ y]$$

e logo, pela Definição 4,  $f$  é uma forma quadrática.

Como o teorema espectral na forma simplificada dada no Teorema 3 diagonaliza uma matriz simétrica, o teorema a seguir permite transformar uma forma quadrática qualquer, em uma forma quadrática sem termos mistos.

**Teorema 5.** *(Teorema dos eixos principais) Considere  $A \in \mathbb{R}^{n \times n}$  uma matriz associada à forma quadrática  $x^T A x$ . Se  $Q$  é uma matriz ortogonal tal que  $Q^T A Q = D$  é diagonal. Então*

$\forall y \in \mathbb{R}^n$  a mudança de variáveis  $x = Qy$  transforma a forma quadrática  $x^T Ax$  em  $y^T Dy$ , e para  $y = [y_1 \ \cdots \ y_n]^T$  e  $\lambda_i$  autovalores de  $A$

$$x^T Ax = y^T Dy = \sum_{i=1}^n \lambda_i y_i^2.$$

*Prova.* Por definição de forma quadrática,  $A$  é simétrica, e pelo Teorema 3 a matriz diagonal  $D \in \mathbb{R}^{n \times n}$  da hipótese é uma matriz com os autovalores  $\lambda_i$  de  $A$  na diagonal principal.

Considere a mudança de variáveis  $x = Qy$ . Logo, substituindo na forma quadrática  $x^T Ax$ , obtemos

$$x^T Ax = (Qy)^T A(Qy) = y^T Q^T A Q y = y^T D y = \sum_{i=1}^n \lambda_i y_i^2$$

que é uma forma quadrática sem termos mistos pois  $D$  é diagonal.  $\square$

O próximo resultado permite determinar limitantes inferior e superior para uma forma quadrática qualquer.

**Teorema 6.** Considere a forma quadrática  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , com  $f(x) = x^T Ax$ . Sejam  $\lambda_n \geq \lambda_1$  o maior e o menor autovalores de  $A$ . Então para todo  $x \in \mathbb{R}^n$

$$\lambda_1 \|x\|^2 \leq x^T Ax \leq \lambda_n \|x\|^2.$$

*Prova.* Como  $A$  é simétrica, o Teorema 3 permite diagonalizar ortogonalmente esta matriz. Logo, existe uma matriz ortogonal  $Q$  tal que  $Q^T A Q = D$ , com

$$D = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix}.$$

Pelo teorema dos eixos principais, Teorema 5, a mudança de variáveis  $x = Qy$  permite escrever

$$x^T Ax = y^T D y. \quad (2)$$

Como  $x = Qy$  obtemos que  $y = Q^T x$ . Mas  $Q$  ortogonal implica em  $Q Q^T = I$ , logo,

$$\|y\|^2 = (Q^T x)(Q^T x) = x^T Q Q^T x = x^T x = \|x\|^2. \quad (3)$$

Seja  $y = [y_1, \cdots, y_n]^T$ , logo, de (2) e (3) podemos escrever

$$x^T Ax = y^T D y = \sum_{i=1}^n \lambda_i y_i^2 \geq \lambda_1 \sum_{i=1}^n y_i^2 = \lambda_1 \|y\|^2 = \lambda_1 \|x\|^2,$$

e

$$x^T Ax = y^T D y = \sum_{i=1}^n \lambda_i y_i^2 \leq \lambda_n \sum_{i=1}^n y_i^2 = \lambda_n \|y\|^2 = \lambda_n \|x\|^2,$$

completando a demonstração.  $\square$

### 3 Problema de otimização restrita

Se uma forma quadrática  $f$  for positiva, isto é, se  $f(x) = x^T A x > 0$  para todo  $x \in \mathbb{R}^n \setminus \{0\}$ , como  $f(0) = 0$ , então o valor mínimo de  $f$  é zero e ocorre na origem. Neste caso não existe maximizador para  $f$ . Analogamente, se  $f(x) = x^T A x < 0$  para todo  $x \in \mathbb{R}^n \setminus \{0\}$ , então o valor máximo de  $f$  é zero e ocorre na origem. Neste caso não existe minimizador para  $f$ . Portanto, para uma forma quadrática  $f$  qualquer, temos a existência de uma solução global de um problema de otimização **irrestrita**. No entanto, o teorema anterior permite resolver problemas de otimização restrita, sem utilizar resultados de Análise. Mais particularmente, podemos resolver o problema de otimização restrita (1).

**Corolário 7.** *Considere a forma quadrática  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , com  $f(x) = x^T A x$ . Seja  $\lambda_n \geq \lambda_1$  o maior e o menor autovalor de  $A$ . Então o valor máximo de  $f$  para  $\|x\| = r$  é  $r^2 \lambda_n$ , e o valor mínimo de  $f$  é  $r^2 \lambda_1$ .*

*Prova.* Pelo Teorema 6, para todo  $x \in \mathbb{R}^n$

$$\lambda_1 \|x\|^2 \leq x^T A x \leq \lambda_n \|x\|^2.$$

Logo, se  $\|x\| = r$ , então vale

$$r^2 \lambda_1 \leq f(x) \leq r^2 \lambda_n.$$

Resta mostrar que os valores  $r^2 \lambda_1$  e  $r^2 \lambda_n$  são atingidos. De fato, considere  $v_1 \in \mathbb{R}^n$  um autovetor associado ao autovalor  $\lambda_1$ , logo, por definição,  $A v_1 = \lambda_1 v_1$ . Assim,

$$f\left(r \frac{v_1}{\|v_1\|}\right) = \left(r \frac{v_1}{\|v_1\|}\right)^T A \left(r \frac{v_1}{\|v_1\|}\right) = r^2 \frac{v_1^T}{\|v_1\|^2} A v_1 = r^2 \frac{v_1^T}{\|v_1\|^2} \lambda_1 v_1 = r^2 \lambda_1 \frac{\|v_1\|^2}{\|v_1\|^2} = r^2 \lambda_1.$$

Analogamente, para  $v_n$  autovetor associado a  $\lambda_n$ , temos

$$f\left(r \frac{v_n}{\|v_n\|}\right) = r^2 \lambda_n.$$

□

Pelo Corolário 7 temos que o valor mínimo de uma forma quadrática  $f$  na fronteira de uma bola unitária é o menor autovalor,  $\lambda_1$ , da matriz associada à forma quadrática  $f$  e, ocorre no autovetor normalizado associado a  $\lambda_1$ . O valor máximo é o maior autovalor,  $\lambda_n$ , da matriz associada à forma quadrática  $f$  e, ocorre no autovetor normalizado associado a  $\lambda_n$ .

**Exemplo 2.** Considere a bola unitária e a forma quadrática  $f(x, y) = 2x^2 - y^2 - xy$  no problema (1). A matriz  $A$  associada a  $f$  tem autovalores  $\lambda_1 \approx 2,081$  e  $\lambda_2 \approx -1,081$ , com autovetores unitários  $v_1 = [-0,98708 \ 0,16018]^T$  e  $v_2 = [-0,1602 \ -0,9871]^T$ , respectivamente. Logo, o ponto que maximiza  $f$  na bola unitária é o ponto  $v_1$ , e o valor máximo correspondente é  $\lambda_1$ . O ponto que minimiza  $f$  na bola unitária é o ponto  $v_2$ , e o valor mínimo correspondente é  $\lambda_2$ . A Figura 11 ilustra este exemplo apresentando os pontos  $w_1 = [v_1, 0] \in \mathbb{R}^3$  e  $w_2 = [v_2, 0] \in \mathbb{R}^3$  cujas duas primeiras coordenadas estão no domínio da  $f$  e representam seus otimizadores, e os respectivos pontos  $u_1 = [v_1, \lambda_1]$  e  $u_2 = [v_2, \lambda_2]$  no gráfico da função objetivo.

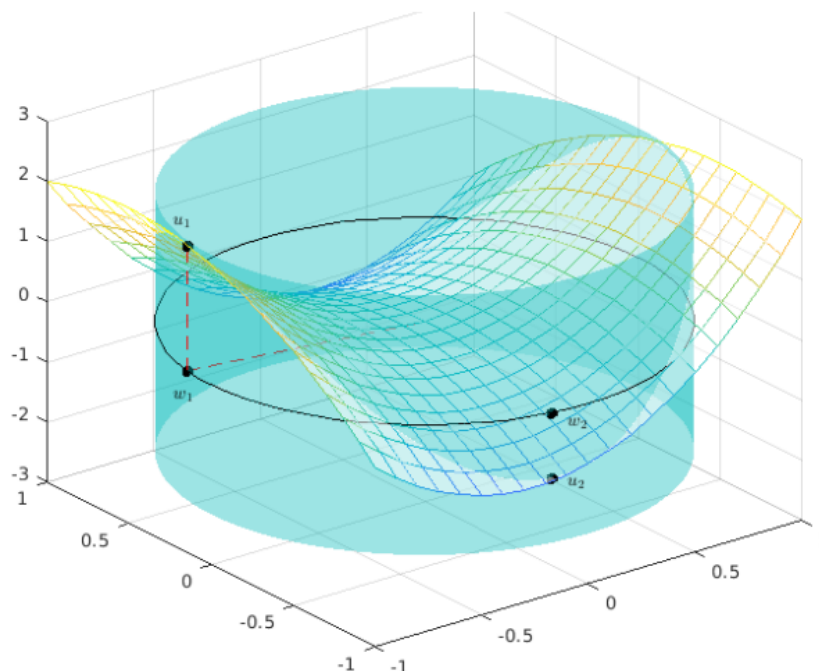


Figura 11: Ilustração do problema (1) para  $f(x, y) = 2x^2 - y^2 - xy$ .

Fonte: Autores.

## Conclusões

Podemos observar que o Corolário 7 trata o problema de otimização restrita (1) utilizando apenas elementos da Álgebra Linear, não obstante à importância da Análise Matemática para trabalhar com problemas gerais de otimização. A partir de resultados clássicos como o teorema espectral e dos eixos principais, foi possível estabelecer a existência de ótimos para o problema restrito a uma bola euclidiana, com função objetivo definida por uma forma quadrática.

## Referências

- HOFFMAN, Kenneth; KUNZE, Ray. Álgebra linear. 2 ed. Rio de Janeiro: Livros Técnicos e Científicos, 1979. 514 p.
- LIMA, E. L. Álgebra Linear. 2 ed. Instituto de Matemática Pura e Aplicada, 1996 (Coleção Matemática Universitária). 357 p.
- POOLE, David. Álgebra Linear. 2 ed., Cengage Learning, São Paulo, 2016. 690 p.
- RIBEIRO, Ademir Alves; KARAS, Elizabeth Wegner. Otimização Contínua: Aspectos Teóricos e Computacionais. São Paulo: Cengage Learning, 2013. 300 p.

## Simplicidade dos grupos alternados

Fernando Dias de Oliveira Heidercheidt  
Universidade Estadual do Oeste do Paraná  
fernando.heidercheidt@unioeste.br

Prof. Dr. Rian Lopes de Lima  
Universidade Estadual do Oeste do Paraná  
rian.lima@unioeste.br

**Resumo:** O grupo alternado  $A_n$  é o subgrupo de  $S_n$  formado pelas permutações pares. Este trabalho se propõe a estudar a simplicidade do grupo alternado  $A_n$ , isto é, estudar os seus subgrupos normais. Para tanto, será analisada a estrutura dos 3-ciclos do grupo alternado.

**Palavras-chave:** Grupo; Grupo simétrico; Grupo alternado; Grupo simples.

### 1 Grupos e subgrupos

**Definição 1.** Seja  $*$  uma operação em um conjunto não vazio  $G$ . O par  $(G, *)$  é chamado de *grupo* se satisfizer as seguintes condições:

1. Associatividade:  $\forall a, b, c \in G, (a * b) * c = a * (b * c)$ ;
2. Existência do elemento neutro:  $\exists e \in G$  tal que  $e * a = a * e = a \forall a \in G$ .
3. Existência do elemento simétrico:  $\forall a \in G \exists a' \in G$  tal que  $a * a' = a' * a = e$ .

Por conveniência, o símbolo da operação do grupo pode ser omitido e os elementos operados são escritos por justaposição.

**Proposição 2.** O conjunto  $S_X = \{f : X \rightarrow X \text{ tal que } f \text{ é bijetiva}\}$  munido com a composição de funções  $\circ$  é um grupo.

Este grupo é chamado de *Grupo Simétrico* de  $X$ . Nosso maior interesse aqui é sobre o conjunto  $X = X_n := \{1, 2, \dots, n\}$ , escrevemos

$$S_n = \{\sigma : \{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, n\} \mid \sigma \text{ é uma bijeção}\}$$

e o chamamos de Grupo Simétrico de  $n$  elementos.

É fácil ver que existem, no máximo,  $n!$  permutações. Desta forma, pode-se escrever uma permutação como um conjunto finito de aplicações em que cada elemento é levado na sua imagem, mas essa imagem (qualquer que seja) sempre é um elemento do domínio, que é levado na sua respectiva imagem e assim, sucessivamente. Daí,

$$k_1 \rightarrow k_2 \rightarrow \dots \rightarrow k_{r-1} \rightarrow k_r \rightarrow k_1$$

até formar um ciclo que volta no elemento  $k_1$ . Notemos ainda que  $\{k_1, k_2, \dots, k_{r-1}, k_r\} \subseteq X_n$ , onde  $r \leq n$ . Agora, podemos definir melhor o que é um  $r$ -ciclo.

**Definição 3.** Dizemos que um ciclo  $\sigma = (k_1 \ k_2 \ \dots \ k_{r-1} \ k_r)$  que permuta  $r$  números inteiros distintos é chamado de  $r$ -ciclo.

Analogamente, o inverso de um  $r$ -ciclo deve percorrer o caminho inverso

$$k_1 \rightarrow k_r \rightarrow k_{r-1} \rightarrow \dots \rightarrow k_2 \rightarrow k_1.$$

Portanto,

$$\sigma = (k_1 \ k_2 \ \dots \ k_{r-1} \ k_r) \Rightarrow \sigma^{-1} = (k_1 \ k_r \ \dots \ k_3 \ k_2).$$

Há um caso particular de um ciclo chamado de transposição, este seria o equivalente a um 2-ciclo, pois permuta dois elementos e fixa todos os demais.

**Exemplo 1.** Sejam  $\sigma = (1567)(2467), \tau = (134)(468) \in S_8$ , temos que

$$\begin{aligned}\sigma\tau^{-1} &= (1567)(2467)((134)(468))^{-1} \\ &= (1567)(2467)(486)(143) \\ &= (18)(243567).\end{aligned}$$

**Definição 4.** Um subconjunto não vazio  $H \subseteq G$  é dito ser um subgrupo de  $G$  se ele é um grupo considerando a restrição de  $*$  a  $H$ .

Usa-se a notação  $H \leq G$  para denotar que  $H$  é subgrupo de  $G$ .

**Exemplo 2.** O conjunto  $\{(1), (123), (132)\}$  é subgrupo de  $S_3$  do exemplo.

**Proposição 5.** Seja  $H$  um subconjunto não vazio de um grupo  $G$ . Se  $xy^{-1} \in H \ \forall x, y \in H$ , então  $H$  é subgrupo de  $G$ .

## 2 Subgrupo normal

**Definição 6.** Seja  $H \leq G$  e  $g \in G$ . Uma *classe lateral à esquerda de  $H$  determinada por  $g$*  é definida pelo conjunto

$$gH := \{gx \mid x \in H\}.$$

Já uma *classe lateral à direita de  $H$  determinada por  $g$*  é dada por

$$Hg := \{xg \mid x \in H\}.$$

**Definição 7.** Um subgrupo  $H$  de  $G$  é dito ser normal se

$$gH = Hg, \ \forall g \in G.$$

A notação que representa um subgrupo normal é  $H \trianglelefteq G$ .

**Proposição 8.** Seja  $H \leq G$  e  $g \in G$ . Temos que  $H \trianglelefteq G$  se, e somente se,  $gHg^{-1} = H \ \forall g \in G$ .

Assim, é interessante perceber que basta provar que  $gHg^{-1} \subset H$  para que  $H$  seja normal a  $G$ .

**Definição 9.** Um grupo  $G \neq \{e\}$  é *simples* se os únicos subgrupos normais são os subgrupos triviais  $\{e\}$  e  $G$ .

### 3 Sinal e Grupo Alternado

**Proposição 10.** *Todo  $r$ -ciclo pode ser escrito como o produto de  $r - 1$  transposições.*

**Corolário 11.** *Toda permutação pode ser escrita como produto de transposições.*

**Definição 12.** Seja  $N_\sigma$  o número de transposições de uma decomposição em transposições de  $\sigma \in S_n$ . Se  $N_\sigma$  é um número par (ou ímpar), diz-se que  $\sigma$  é uma permutação par (ou ímpar).

**Proposição 13.** *Uma permutação ou é par ou é ímpar.*

**Definição 14.** O sinal de  $\sigma \in S_n$  ( $\text{sgn}(\sigma)$ ) é 1 se  $\sigma$  for par e é  $-1$  se  $\sigma$  for ímpar.

**Exemplo 3.** Determinemos o sinal de  $\sigma = (1234)(123)$

Escrevendo

$$\sigma = (14)(13)(12)(13)(12)$$

podemos afirmar que  $\text{sgn}(\sigma) = -1$ , pois  $N_\sigma = 5$

**Proposição 15.** *Considerando  $\sigma, \tau \in S_n$ , vale que:*

i.  $\text{sgn}(\sigma\tau) = \text{sgn}(\sigma)\text{sgn}(\tau),$

ii.  $\text{sgn}(\sigma^{-1}) = \text{sgn}(\sigma).$

**Proposição 16.**  $A_n := \{\sigma \mid \text{sgn}(\sigma) = 1\} < S_n.$

Este subgrupo  $A_n$  é chamado de *subgrupo alternado* de  $S_n$ .

**Exemplo 4.** O subgrupo  $\{(1), (123), (132)\}$  do exemplo 2 é o grupo alternado  $A_3$ .

**Exemplo 5.** O grupo alternado

$$A_4 = \{(1), (12)(34), (13)(24), (14)(23), (123), (132), (134), (143), (234), (243), (124), (142)\}$$

é um subgrupo de  $S_4$ .

### 4 Simplicidade de $A_n$

A partir de agora, apresentaremos o estudo da estrutura dos 3-ciclos em  $A_n$  como um caminho para decidir sobre a simplicidade do grupo alternado.

**Lema 17.** *Para  $n \geq 3$ ,  $A_n$  contém todos os 3-ciclos.*

**Lema 18.** *Para  $n \geq 3$ ,  $A_n$  é gerado pelos 3-ciclos.*

**Proposição 19.**  $A_n$  é gerado por  $n - 2$  3-ciclos da forma  $(abi)$ , onde  $a \neq b$ ,  $i \neq a$  e  $i \neq b$ .

**Proposição 20.** *Seja  $N \neq \{(1)\}$  um subgrupo normal de  $A_n$ ,  $n \geq 3$ . Se  $N$  contém um 3-ciclo, então  $N$  é o próprio grupo alternado  $A_n$ .*

No que concerne a simplicidade de  $A_n$ , temos:

- Temos que  $A_1 = A_2 = \{(1)\}$ , que não é simples.
- O grupo  $A_3 = \{(1), (123), (132)\}$  é um  $p$ -grupo e dessa forma é simples.
- O grupo de Klein  $K_4 = \{(1), (12)(34), (13)(24), (14)(23)\}$  é subgrupo normal de  $A_4$ . Logo, segue que  $A_4$  não é simples.

Já para  $n \geq 5$  temos um padrão para a simplicidade dos grupos alternados:

**Proposição 21.** *O grupo alternado  $A_n$  é simples para  $n \geq 5$ .*

## Referências

- DOMINGUES, H., IEZZI, G. **Álgebra Moderna**, 4<sup>a</sup> ed. São Paulo: Atual, 2003.
- GARCIA, A., IEQUAIN, Y. **Elementos de álgebra**, 6<sup>a</sup> ed. Rio de Janeiro: IMPA, 2013.
- JACOBSON, N. **Basic Algebra I**, 2nd ed. New York: Dover Publications, 1985.
- MARTIN, P. A. **Grupos, Corpos e Teoria de Galois**. São Paulo: Editora Livraria da Física, 2010.
- ROSE, J. S. **A course on group theory**. New York: Dover Publications, 1978.