

UNIVERSIDADE ESTADUAL DE CAMPINAS
SISTEMA DE BIBLIOTECAS DA UNICAMP
REPOSITÓRIO DA PRODUÇÃO CIENTÍFICA E INTELECTUAL DA UNICAMP

Versão do arquivo anexado / Version of attached file:

Versão do Editor / Published Version

Mais informações no site da editora / Further information on publisher's website:

<https://pmo.sbm.org.br/volumes-anteriores/>

DOI: 10.21711/2319023x2023/pmo1104

Direitos autorais / Publisher's copyright statement:

©2023 by Sociedade Brasileira de Matemática. All rights reserved.

DIRETORIA DE TRATAMENTO DA INFORMAÇÃO

Cidade Universitária Zeferino Vaz Barão Geraldo

CEP 13083-970 – Campinas SP

Fone: (19) 3521-6493

<http://www.repositorio.unicamp.br>

Animações para o ensino de Matemática usando o Manim–Python

Eric Satoshi Suzuki Kishimoto 

Vitor Rafael Coluci 

Resumo

Animação refere-se ao processo de dar movimento a objetos inanimados. Animações têm sido usadas como facilitadoras de aprendizagem de Matemática tanto em sala de aula como no estudo individual. Diversos *softwares* têm permitido a criação de animações cada vez mais sofisticadas. Dentre eles está o **Manim**, uma biblioteca do **Python** que dá movimento a fórmulas matemáticas, figuras geométricas, textos e gráficos e possibilita apresentar visualmente conteúdos matemáticos de forma mais atrativa. Neste trabalho, usamos o **Manim** para criar animações de temas de Matemática abordados no Ensino Médio e Superior. As animações produzidas envolveram os temas de números irracionais, geometria, retas, cônicas, sistemas lineares, interpolação polinomial, conjuntos, funções, progressões, combinatória e probabilidade. Além das animações, elaboramos um manual em português do **Python/Manim** que fornece os principais conceitos de programação para que interessados iniciantes possam desenvolver suas próprias animações. Tanto as animações produzidas como o processo de criá-las podem servir para atividades complementares para o ensino de Matemática e facilitadoras de seu aprendizado.

Palavras-chave: animação; Python; pensamento computacional

Abstract

Animation is the process of giving motion to lifeless objects. Animations have been employed as learning aids for Mathematics both in the classroom and in individual study. A variety of programs has made possible to create animations that are getting more and more complex. One of them is **Manim**, a library of **Python**, which brings motion to mathematical formulas, geometric figures, texts, and plots and allows for the more appealing visual presentation of mathematical knowledge. In this work, we use the **Manim** to make animations of Mathematics topics covered in High School and Higher Education. Irrational numbers, geometry, lines, conics, linear systems, polynomial interpolation, sets, functions, progressions, combinatorics, and probability were among the topics covered in our animations. Along with the animations, we also produced a guidebook of **Python/Manim** in Portuguese that explains the fundamental programming concepts so that interested novices can make their own animations. The animations themselves as well as the process of making them can both be used in supplemental activities and as learning aids for Mathematics.

Keywords: animation; Python; computational thinking

¹Apoiado pelo CNPq bolsa 0220036212472856.

1. Introdução

O conhecimento científico é primordial para alcançar o desenvolvimento e a justiça social, e a base para isso é a Matemática. Apesar desse papel fundamental, o desempenho dos estudantes brasileiros em Matemática tem evoluído pouco. O Brasil ficou em 58º lugar (de 64 países) em Matemática no Pisa/2018 (*Programme for International Student Assessment*) com uma pontuação de 384 [15], valor 22% menor que a média da Organização para a Cooperação e Desenvolvimento Econômico [19]. Além disso, do total de estudantes, apenas 32% atingiram o resultado mínimo em Matemática. Os resultados do Brasil no Pisa nas versões anteriores mostram que não houve uma melhora significativa no desempenho em Matemática nos últimos 15 anos (377 (2015), 389 (2012), 386 (2009), 370 (2006) e 356 (2003) [15]).

Várias iniciativas têm sido realizadas para facilitar a aprendizagem de Matemática. Uma dessas iniciativas envolve o uso de *softwares* para apresentar conteúdos e complementar as aulas. Por exemplo, o *software* GeoGebra tem sido usado para explorar conceitos de geometria hiperbólica [21], equações algébricas [17], elipse [22], otimização [12], geometria [6, 16, 2] e funções [1]. No entanto, a maioria dos tópicos de Matemática ainda é apresentada aos estudantes num texto estático, tradicionalmente por meio de livros ou alguma outra forma textual. Enquanto essas apresentações estáticas de um tópico como, por exemplo, a representação de uma matriz, são úteis para informar o conteúdo, o uso de animações é mais indicado para apresentar e explicar os processos da construção do conteúdo. Por exemplo, Taylor *et al.* mostraram que a apresentação de assuntos como adição e multiplicação de matrizes é mais efetiva em termos de aprendizagem por meio de animações do que quando apresentada de forma estática [23]. No trabalho, a adição de duas matrizes 2×2 é animada fazendo-se que os números de cada matriz, em suas respectivas posições, movam-se até a posição na matriz resultante, acrescentando-se o sinal de soma e, finalmente, indicando o resultado.

Animações têm se mostrado elementos motivadores de aprendizagem de Matemática. Por exemplo, Moraes *et al.* usaram o Geogebra para explorar conceitos de geometria analítica (ponto, reta e circunferência) com estudantes de graduação em Matemática Aplicada [14]. O Geogebra permite uma animação interativa, já que o usuário pode interagir com os objetos e os elementos da animação. Os estudantes realizaram a construção e animação do movimento de um carro em linha reta. Essa atividade levou a uma maior interação, cooperação e aprendizagem. Monzo e Gravina usaram também animações interativas para introduzir o conceito de funções de variáveis complexas para estudantes do Ensino Médio [13]. Nesse caso, o uso de animações desencadeou um processo de aprendizagem com autonomia, contemplando a conexão entre geometria e álgebra, e permitiu a exploração do conceito, geralmente inibida num texto estático.

A origem da palavra animação é latina, de *animatio*, que significa, ser animado. Deriva também da palavra *anima*, que significa *alma* ou *sopro vital*. Assim, o processo de animar refere-se ao de dar vida a algo sem vida ou sem movimento. Animações de elementos matemáticos como fórmulas, matrizes e figuras geométricas tiveram um grande desenvolvimento nos últimos 40 anos. Por exemplo, a série “O Universo Mecânico” criada por David Goodstein em 1985 usa muito dessas animações [10]. Animações mais sofisticadas e mais fáceis de serem produzidos têm sido possíveis com o avanço de ferramentas de animação. Especificamente, para animações de elementos matemáticos, uma ferramenta que tem evoluído muito nos últimos anos é o **Manim** (*Mathematical Animation Engine*).

O **Manim** é uma biblioteca gratuita do **Python**, criada por Grant Sanderson, mantenedor do projeto [3blue1brown](#). Capaz de produzir uma grande variedade de animações envolvendo fórmulas,

manipulações algébricas, gráficos e objetos geométricos, o **Manim** tem possibilitado apresentar conteúdos matemáticos visualmente atrativos que vão desde a demonstração geométrica do **teorema de Pitágoras** até a explicação do que são **séries de Fourier** e o uso da Matemática na teoria de erros usada em Física experimental [4]. Além disso, o **Manim** atualmente possui uma comunidade ativa que mantém o site <https://www.manim.community/> com documentação, tutoriais e exemplos.

Neste trabalho, desenvolvemos animações matemáticas com o **Manim** para complementar temas de Matemática abordados no Ensino Médio e Superior. Além das animações produzidas, o trabalho visa também divulgar e disseminar as potencialidades dessa ferramenta no ensino de Matemática. Para isso, elaboramos um manual em português de introdução ao **Python** e ao **Manim** para iniciantes em programação computacional, a fim de que, tanto professores como estudantes, possam aprender a ferramenta e desenvolver suas próprias animações. O manual está disponível gratuitamente em duas versões: uma **estática**, onde não é possível executar os códigos; e uma **interativa**, onde se podem executar os códigos em **Python** lá apresentados no próprio navegador, sem a necessidade de instalação local de qualquer programa ou biblioteca.

Para ilustrar o processo de criação de uma animação com o **Manim**, apresentamos na seção 2 detalhes de tal processo para obter o comprimento da diagonal de um quadrado. A descrição das animações produzidas neste trabalho está apresentada na seção 3. Finalmente, as considerações finais são apresentadas na seção 4.

2. Exemplificando com a determinação da diagonal do quadrado

Nesta seção ilustraremos algumas possibilidades de animação com o **Manim**. Para isso, mostraremos um código para produzir uma animação onde é feita a determinação do valor da diagonal de um quadrado de lado a . A escolha desse tópico é interessante pois, apesar de a álgebra ser simples, a animação envolve elementos que podem ser usados em muitas outras animações com o **Manim**. Por exemplo, a animação envolve equações, movimentação e manipulação algébrica de equações, objetos geométricos (quadrado e triângulo) e o uso de cores. Usamos aqui uma programação estruturada para facilitar o acompanhamento do funcionamento do código, principalmente para iniciantes. Uma versão do código com uma programação orientada a objetos pode ser encontrada no manual.

Antes da implementação do código da animação, recomendamos elaborar um roteiro das etapas da animação. Isso permite uma organização e otimização do trabalho de implementação. Um possível roteiro com as etapas da animação está mostrado a seguir (Fig. 1).

1. A animação começa com um quadrado aparecendo na tela.
2. Depois disso, o valor do lado (a) é mostrado próximo a dois lados consecutivos do quadrado.
3. Em seguida, a diagonal do quadrado aparece na tela com uma cor diferente da do quadrado.
4. Próximo à diagonal, aparece então o texto $x = ?$, para indicar que queremos determinar o comprimento da diagonal.
5. O passo seguinte é o de destacar o triângulo formado por dois lados consecutivos do quadrado e pela diagonal, indicando que é um triângulo retângulo (mostrando que o ângulo entre os lados é reto). O destaque é feito desenhando o triângulo com outra cor.

6. Com a informação de que é um triângulo retângulo, é feita a manipulação algébrica: $x^2 = a^2 + a^2 \rightarrow x^2 = 2a^2 \rightarrow x = \sqrt{2a^2} \rightarrow x = \sqrt{2}a$.
7. Finalmente, após essa manipulação, a expressão $x = \sqrt{2}a$ é movida para a região próxima à diagonal.

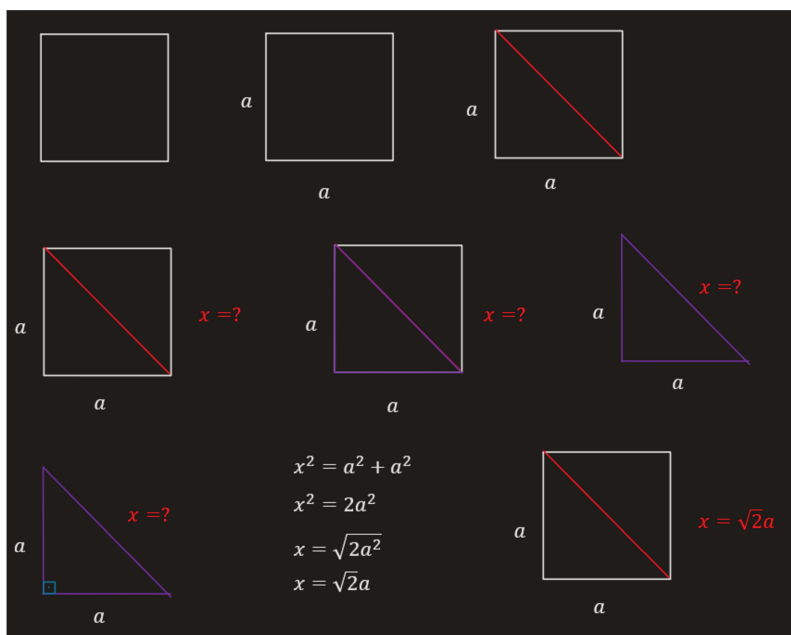


Figura 1: Visualização do roteiro para a animação onde a diagonal do quadrado é calculada.

Com o roteiro pronto, o próximo passo é codificar a animação. Para criar uma animação com o **Manim**, é necessário primeiramente criar uma classe. Em programação, uma classe é definida como uma abstração de um objeto. Essa classe herda da classe **Scene** do **Manim**. Dentro dela, é necessário criar um método chamado **construct**. Em programação, um método é definido como uma função dentro de uma classe. No método **construct** é onde se criam os objetos como fórmulas e gráficos, e onde eles são mostrados e animados. Alguns tipos de animações são mostrar, remover, mover, desenhar o objeto na tela, dentre outros.

O código a seguir produz a animação em que se obtém a diagonal de um quadrado de lado a , aplicando-se o teorema de Pitágoras. A primeira parte do código é usada para construir e posicionar os objetos que serão usados na animação: quadrado (**Square**), triângulo (**Polygon**), segmentos de reta (**Line**), pontos (**Dot**) e pontos de referência, textos e equações (na linguagem **LATEX** com **MathTex**). Quando um objeto é criado pelo **Manim**, ele é posicionado, como padrão, no centro da tela $([0,0,0])$. O posicionamento dos objetos em lugares é feito por meio dos comandos **move_to** e **next_to**, usando vetores de direção como o **RIGHT** $= [1,0,0]$, **LEFT** $= [-1,0,0]$, **UP** $= [0,1,0]$ e **DOWN** $= [0,-1,0]$. O tamanho dos objetos é ajustado pelo comando **scale()**. As cores dos objetos são definidas com o comando **set_color()**. Várias cores já estão predefinidas no **Manim** como **WHITE**, **YELLOW**, **RED** e **PURPLE**, usadas na presente animação. O **Manim** também oferece a opção de agrupar objetos

com o comando `VGroup`. No nosso caso, usamos esse recurso para criar um objeto que representasse o símbolo do ângulo reto, formado por um quadrado e um ponto no seu centro.

A segunda parte é destinada para fazer a animação dos objetos propriamente dita. Isso é feito basicamente por meio do comando `self.play()` com suas diferentes opções. Por exemplo, a opção `FadeIn` faz o objeto aparecer gradualmente na tela e o comando `Write` produz o efeito de escrever/desenhar o objeto na tela. Efeitos mais sofisticados também são possíveis como, por exemplo, transformação de um objeto em outro com a opção `ReplacementTransform`. Assim, o comando `self.play(ReplacementTransform(x_nome.copy(), equacao1[0]))` transforma uma cópia do texto `x_nome` (`x_nome.copy() = "x = ?"`) na primeira componente da equação 1 (`equacao1[0] = "x2"`). O comando `TransformMatchingTex` tem um efeito similar nesse caso, pois ele transforma um texto em $\text{\LaTeX}$ num outro texto em $\text{\LaTeX}$, contanto que os dois textos coincidam. Pausas na animação são produzidas com o comando `self.wait()`, o qual cria uma pausa de 1 s.

Finalmente, o vídeo da animação é produzido com o comando `manim -qm -v DiagonalQuadrado` executado na linha de comando. A opção `-qm` indica uma renderização de média qualidade (1280×720 com 30 quadros por segundo). Outras opções para o comando `manim` estão disponíveis e podem ser encontradas em <https://www.manim.community/>.

```

1 from manim import *
2 # criação da classe
3 class DiagonalQuadrado(Scene):
4     # criação do método construct
5     def construct(self):
6         #-----
7         # Construção dos objetos geométricos
8         #-----
9
10        # quadrado
11        quadrado = Square(side_length=4)\
12            .set_color(WHITE)\
13            .move_to(2*LEFT)
14
15        # pontos da diagonal do quadrado
16        pontos_diagonal = (
17            quadrado.point_from_proportion(0.25),
18            quadrado.point_from_proportion(0.75)
19        )
20        # segmento de reta que representa a diagonal
21        diagonal = Line(*pontos_diagonal).set_color(RED)
22        %
23        %
24        # vértices do quadrado com os quais será desenhado o triângulo
25        cantos_triângulo = (
26            quadrado.point_from_proportion(0.25),
27            quadrado.point_from_proportion(0.5),
28            quadrado.point_from_proportion(0.75)
29        )
30        # triângulo
31        triângulo = Polygon(*cantos_triângulo).set_color(PURPLE)
32
33        # posição onde será colocado um ponto para representar o
34        # ângulo reto
35        ponto_angulo_retangulo = quadrado.point_from_proportion(0.5)\
36            + 0.5*0.4*UP\
37            + 0.5*0.4*RIGHT
38
39        # quadrado com um ponto no centro para representar o
40        # ângulo reto

```

```

41 angulo_retangulo = VGroup(
42     Square(side_length=0.4)\
43     .move_to(ponto_angulo_retangulo)
44     .set_color(BLUE),
45     Dot(ponto_angulo_retangulo).scale(0.15*4)
46 )
47 #-----
48 # Textos
49 #-----
50
51 # nome dos lados do quadrado
52 lados = VGroup(
53     MathTex('a').next_to(quadrado, direction=LEFT, buff=0.5),
54     MathTex('a').next_to(quadrado, direction=DOWN, buff=0.5)
55 ).set_color(WHITE)
56 %
57 # nome para a diagonal desconhecida
58 x_nome = MathTex('x=', '?').next_to(quadrado, direction=RIGHT, buff=0.5).
set_color(RED)
59
60 #-----
61 # Equações
62 #-----
63 equacao1 = MathTex('x^2', '=', 'a^2', '+', 'a^2').move_to(3*RIGHT)
64 equacao2 = MathTex('x^2=2a^2').move_to(3*RIGHT)
65 equacao3 = MathTex('x=\sqrt{2a^2}').move_to(3*RIGHT)
66 equacao4 = MathTex('x=', '\sqrt{2}a').move_to(3*RIGHT)
67 resultado = MathTex('x=', '\sqrt{2}a')\
68     .move_to(triangulo.get_center() + 0.5*UP + 0.8*RIGHT)\
69     .set_color(RED)
70
71 #-----
72 # Animações
73 #-----
74 %
75 # 1) desenha o quadrado
76 self.play(Write(quadrado))
77 # 2) mostra o nome dos lados
78 self.play(Write(lados))
79 # 3) desenha a diagonal e mostra seu valor desconhecido
80 self.play(Write(diagonal), Write(x_nome))
81 # 4) representa o triângulo
82 self.play(Write(triangulo))
83 # 5) remove o quadrado da cena
84 self.play(FadeOut(quadrado))
85 # 6) move o valor desconhecido da diagonal
86 self.play(x_nome.animate.move_to(triangulo.get_center() + 0.5*UP + 0.5*
RIGHT))
87 # 7) mostra o símbolo do ângulo reto
88 self.play(Write(angulo_retangulo))
89 # 8) cria equação que relaciona os lados com a diagonal
90 self.play(
91     ReplacementTransform(x_nome.copy(), equacao1[0]),
92     ReplacementTransform(lados[0].copy(), equacao1[2]),
93     ReplacementTransform(lados[1].copy(), equacao1[4]),
94     t=3
95 )
96 self.play(FadeIn(equacao1[1], equacao1[3]))
97 # aguarda o tempo de 1 s
98 self.wait()

```

```

99 # 9) realiza manipulação algébrica  $a^2 + a^2 = 2a^2$ 
00 self.play(TransformMatchingTex(equacao1, equacao2), t=3)
01 self.wait()
02 # 10) toma a raiz quadrada dos dois lados da equação
03 self.play(TransformMatchingTex(equacao2, equacao3), t=3)
04 self.wait()
05 # 11) obtém o valor de  $x = \sqrt{2}a$ 
06 self.play(TransformMatchingTex(equacao3, equacao4), t=3)
07 self.wait()
08 # 12) apresenta o resultado final junto à diagonal
09 self.play(
10     TransformMatchingTex(x_nome, resultado),
11     ReplacementTransform(equacao4[1].copy(), resultado[1])
12 )
13 self.wait()
14 # fim

```

3. Animações como complementos para o Ensino de Matemática

Apresentamos a seguir 14 animações produzidas com o **Manim** sobre diferentes temas da Matemática do Ensino Médio e Superior. Uma breve descrição de cada animação é apresentada, juntamente com a duração de cada uma delas (em minutos). São animações curtas, com duração inferior a 7 minutos, e que estão disponíveis no YouTube, no [canal](#) do espaço de ensino e aprendizagem Explora [7]. O acesso direto a cada uma delas pode ser feito a seguir, clicando no título de cada uma. Os códigos das animações podem ser obtidos no [github aqui](#).

A escolha dos conteúdos das animações foi baseada em aulas, livros e sítios como o livro de Elon Lages Lima *et. al* [11] e o sítio “Derivando a Matemática” [8]. Esse sítio foi produzido no projeto de iniciação científica de Stephanie G. Nietto e coordenado pelo Prof. Ricardo M. Martins (IMECC/UNICAMP) e tem como objetivo apresentar temas de Matemática de forma contextualizada, dinâmica e divertida. O sítio apresenta temas de geometria, cálculo, álgebra linear, Física e também curiosidades. Dentro dos temas, são apresentados materiais interativos produzidos no Geogebra, vídeos e demonstrações matemáticas. No entanto, as demonstrações são geralmente apresentadas de forma estática, em textos, como nos livros tradicionais.

1. *O número $\sqrt{2}$* (3:11): Essa animação apresenta a origem geométrica do número $\sqrt{2}$ como sendo o comprimento da diagonal de um quadrado de lado unitário. Para isso, apresentamos inicialmente o teorema de Pitágoras. Em seguida, a demonstração geométrica do teorema é animada usando quadrados e triângulos (Fig. 2) e explorando a movimentação desses objetos na tela e manipulações algébricas envolvendo a equação $a^2 = b^2 + c^2$. Finalmente, a $\sqrt{2}$ é obtida para o caso particular de um quadrado de lado unitário.

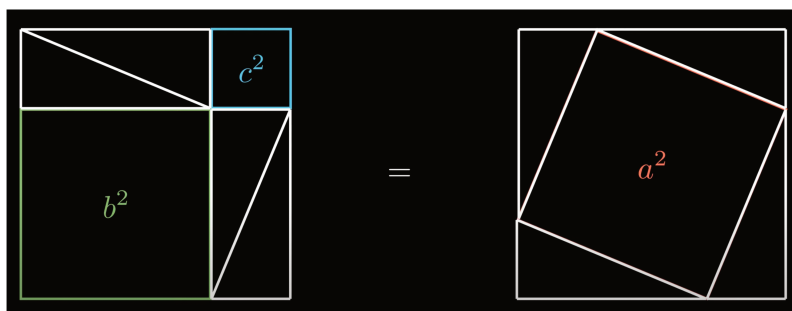


Figura 2: Cena da animação da $\sqrt{2}$ mostrando quadrados, retângulos e triângulos usados na demonstração geométrica do teorema de Pitágoras.

2. *O número π* (1:52): Nessa animação, o número π é obtido a partir de uma abordagem geométrica usando polígonos inscritos numa circunferência. A animação começa com um octógono regular inscrito numa circunferência de raio $r = 1/2$, de onde se deduz a relação de igualdade entre o seno da metade do ângulo central (2θ) com o valor do lado do octógono. Em seguida, obtém-se a relação entre θ , π e o número de lados n do polígono inscrito, $\theta = \pi/n$, e valor do perímetro do polígono $n \sin(\pi/n)$. A animação então apresenta diversos polígonos inscritos em ordem crescente do número de lados e o valor correspondente do perímetro (Fig. 3). O perímetro vai então convergindo para $2\pi r$ que, no nosso caso, vale exatamente π pois $r = 1/2$.

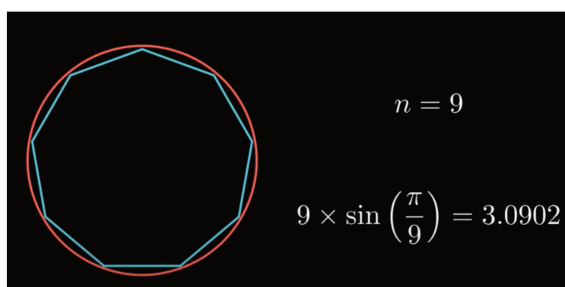


Figura 3: Cena da animação sobre a número π onde um polígono inscrito na circunferência é mostrado juntamente com a aproximação do valor de π obtida a partir do perímetro.

3. *Pontos, Retas e Cônicas* (6:25): A animação mostra a construção e propriedades de curvas no plano cartesiano, usando como exemplos a reta e as cônicas. Para isso, a parte inicial da animação é usada para apresentar como se posicionam pontos no plano cartesiano, primeiramente um único ponto e depois um conjunto de pontos. É indicado então que um conjunto de pontos pode obedecer relações entre suas coordenadas. Isso é feito para introduzir a equação da reta e explicar os coeficientes angular e linear. Em seguida, o conceito é estendido para outras curvas como a elipse, parábola e hipérbole. Para cada uma delas são apresentadas suas propriedades geométricas e equações. Por exemplo, é feita a animação de que um ponto P da elipse satisfaz a relação $d(P, F_1) + d(P, F_2) = 2a$, onde F_1 e F_2 são os focos, a é o semi-eixo maior e $d(A, B)$ é a distância entre os pontos A e B (Fig. 4).

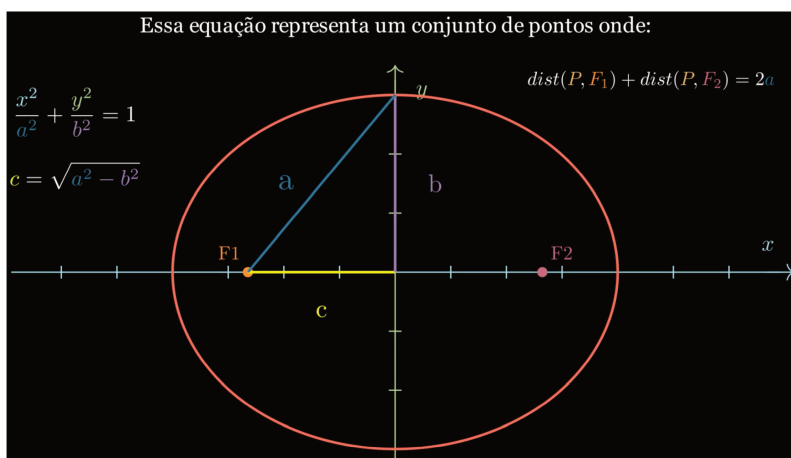


Figura 4: Cena da animação sobre pontos, retas e cônicas. Nesta cena, uma elipse é apresentada na tela com suas equações e sua propriedade geométrica.

4. *Derivação da Equação da Elipse* (2:39): Nessa animação é ilustrada a dedução da equação da elipse na sua forma reduzida $x^2/a^2 + y^2/b^2 = 1$ a partir da condição geométrica $d(P, F_1) + d(P, F_2) = 2a$. Para isso, as manipulações algébricas necessárias são animadas – como, por exemplo, a eliminação da raiz quadrada, que aparece nos termos envolvendo as distâncias, é feita elevando-se ao quadrado os dois lados da equação.
5. *Derivação da Equação da Parábola* (2:30): Animação que ilustra a dedução da parábola na sua forma reduzida $y = (1/4p)x^2$ a partir da condição geométrica $d(P, F) = d(P, r)$, seguindo os mesmos passos usados na animação da elipse.
6. *Derivação da Equação da Hipérbole* (2:35): Animação que ilustra a dedução da hipérbole na sua forma reduzida $x^2/a^2 - y^2/b^2 = 1$ a partir da condição geométrica $|d(P, F_1) - d(P, F_2)| = 2a$, seguindo os mesmos passos usados na animação da elipse e da parábola (Fig. 5).

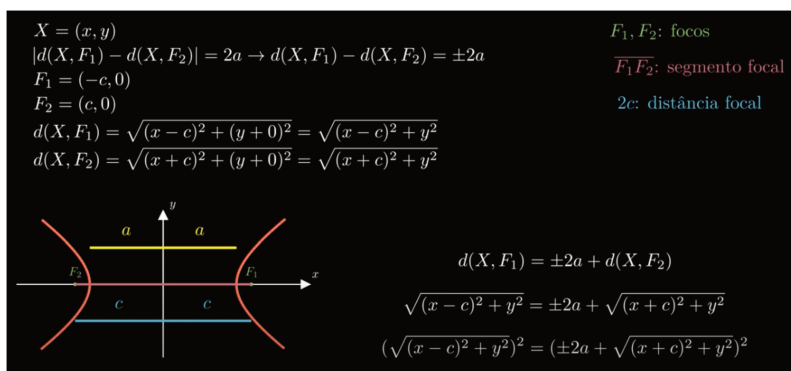


Figura 5: Cena da animação sobre a dedução da equação cartesiana da hipérbole a partir de sua propriedade geométrica.

7. *Lei dos Senos* (4:31): Nessa animação é deduzida a lei dos senos, usando o conceito de produto vetorial. A animação começa apresentando a lei dos senos ao lado de um triângulo inscrito numa circunferência de raio R . Em seguida, são apresentadas algumas propriedades do produto vetorial, principalmente a que $\|\vec{u} \times \vec{v}\| = \|\vec{u}\| \|\vec{v}\| \sin \theta$. A relação do produto vetorial com a área de um paralelogramo é então deduzida. Finalmente, usando o produto vetorial entre os vetores que compõem um triângulo ($\vec{a}$, $\vec{b}$ e $\vec{c}$), a lei dos senos é deduzida calculando-se a área desse triângulo por meio de animações das manipulações algébricas envolvidas.
8. *Sistemas Lineares e Interpolação Polinomial* (2:45): Essa animação foi feita para ilustrar uma aplicação de sistemas de equações lineares no problema de interpolação polinomial – determinação dos coeficientes de um polinômio que passa por pontos do plano cartesiano. Na animação foi usada a parábola $y(x) = a_0 + a_1x + a_2x^2$ como o polinômio interpolante. A partir de 3 pontos do plano cartesiano, a construção do sistema linear foi animada, substituindo-se os valores de x e y dos pontos na expressão da parábola. Após isso, a solução do sistema é apresentada para se mostrar a expressão da parábola obtida. Finalmente, um gráfico é desenhado na tela contendo os pontos e a parábola obtida (Fig. 6).

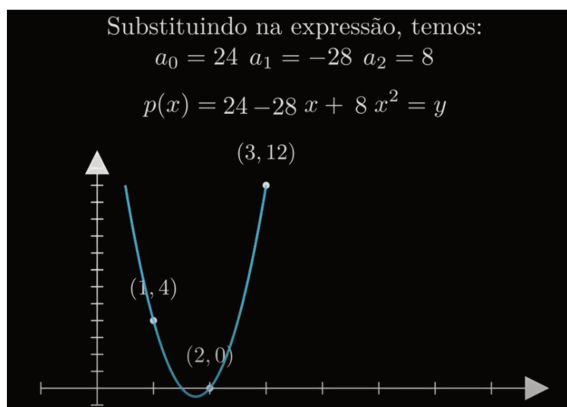


Figura 6: Cena da animação sobre sistemas lineares e interpolação polinomial. Nesta cena, a parábola que interpola um conjunto de 3 pontos é desenhada na tela.

9. *Conjuntos* (5:44): Uma introdução ao conceito de conjunto é apresentada nessa animação. Inicialmente apresentamos um conjunto na forma de um diagrama de Venn para depois indicar que ele pode ser representado por meio de uma lista de elementos ou de uma regra que todos os elementos respeitem. Em seguida, algumas relações entre conjuntos como a de *contido* e *contém* são animadas, usando os diagramas que os representam. Após isso, as relações de *pertence* e *não pertence* entre elementos e conjuntos são apresentadas. Finalmente, as operações entre conjuntos como diferença, interseção e união são apresentadas.
10. *Números Naturais e Inteiros* (4:29): Nessa animação são introduzidos os números naturais e inteiros. Os números naturais são apresentados primeiro, ilustrando o uso deles na contagem de objetos, seguida da representação na forma de diagrama e no conjunto $\mathbb{N} = \{0, 1, 2, 3, \dots\}$. Em seguida, apresentamos as operações aritméticas entre eles (+, -, ×) e as relações (>, <, =). Após isso, os números inteiros são apresentados para então representá-los na reta. O movimento

pela reta é então animado usando um carro que se desloca ao longo da reta, com suas posições centradas nos números inteiros ali indicados.

11. *Números Reais* (2:24): Animação que apresenta o conjunto dos números reais. Inicialmente a animação mostra que ele é formado pela união do conjunto dos números racionais com o dos irracionais, explicando em seguida como são definidos esses números. Essa união é então ilustrada por meio de um diagrama de Venn. O conjunto dos reais é então representado na forma da reta dos reais. Para isso, uma reta horizontal é desenhada na tela e uma flecha, que indica um número específico na reta, é movimentada ao longo da reta (Fig. 7). Essa movimentação visa ilustrar a possibilidade de números “quebrados” que ficam entre os números inteiros. Finalmente, com a reta dos reais, apresentamos o conceito de intervalo, tanto graficamente como textualmente nas formas $I = \{x \in \mathbb{R}, -1 \leq x \leq 1\}$ e $I = [-1, 1]$.

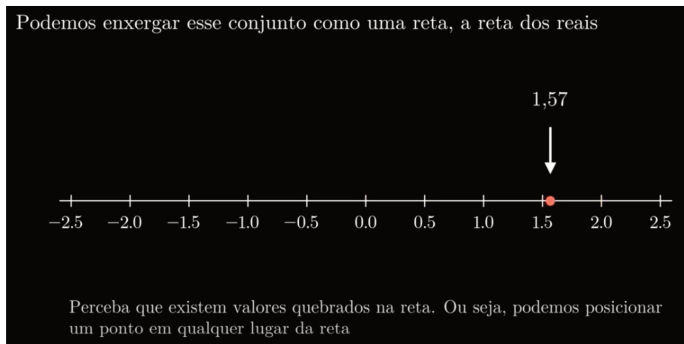


Figura 7: Cena da animação sobre os números reais. Na cena, os números reais são representados na reta. A movimentação da flecha ao longo da reta ilustra o contínuo dos números reais.

12. *Funções* (5:10): Essa animação mostra intuitivamente o conceito de função por meio de gráficos no plano cartesiano e por meio de diagramas de Venn. Mostra também diversos tipos de funções como a quadrática, cúbica, exponencial, logarítmica, senoidal, polinomial, além das combinadas e compostas. A animação começa com a representação de uma função como sendo uma caixa que transforma um número em outro. Em seguida, a função é representada na forma de um gráfico. Diagramas de Venn são usados para indicar os conceitos de domínio e contradomínio, ilustrados também na forma gráfica. Os diagramas de Venn e o gráfico são usados para mostrar situações onde as relações entre números não são consideradas funções. Finalmente, a animação termina apresentando o gráfico das funções $f(x) = x^2$, $f(x) = x^3$, $f(x) = 2^x$, $f(x) = \log_2(x)$, $f(x) = \sin x$, $f(x) = x^3 - x^2 - x + 1$, $f(x) = \log_2(x) + \sin x$ e $f(x) = \sin(x^2 + 1)$ (Fig. 8).

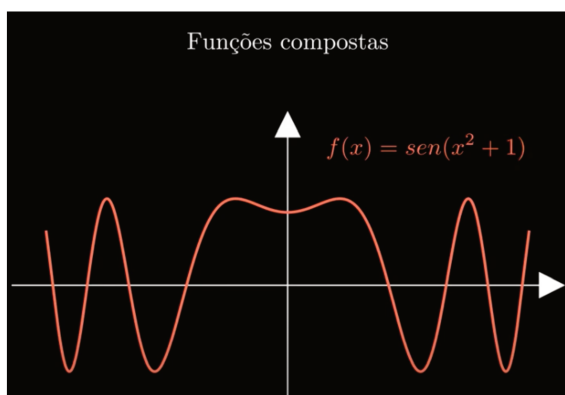


Figura 8: Cena da animação sobre funções onde uma função composta é desenhada na tela.

13. *Progressões* (4:24): Nessa animação mostramos o conceito de progressão aritmética e geométrica de forma gráfica e intuitiva. Inicialmente é apresentada a progressão aritmética usando um conjunto de quadrados de lados que vão aumentando de tamanho. O tamanho do lado do quadrado é então escrito como a progressão $f(n) = 1 + 2(n - 1)$, para então depois chegar na forma mais geral $a_n = a_1 + (n - 1)r$, onde a_j são os termos da progressão, n é a posição do termo na progressão ($n = 1, 2, \dots$) e r é a razão da progressão. Em seguida, usamos a forma de um trapézio e o cálculo de sua área para determinarmos a fórmula da soma dos termos de uma progressão aritmética. O uso dessa fórmula é então exemplificado para se obter a soma dos primeiros 50 termos de $f(n) = 1 + 2(n - 1)$. Finalmente, a progressão geométrica é apresentada, seguindo a mesma estratégia da progressão aritmética mas agora usando um padrão formado por triângulos como ilustração.
14. *Combinatória e Probabilidade* (4:23): Animação que mostra os conceitos de combinatória e probabilidade, com alguns exemplos de permutações e combinações. O exemplo de permutação usado foi o de anagramas que podem ser formados com a palavra CALOR. O exemplo de combinações foi o de se encontrar o número de combinações que podem ser obtidas para uma casquinha de sorvete contendo 2 sabores numa sorveteria que vende 6 sabores diferentes (Fig. 9). Em seguida, a animação apresenta o conceito de probabilidade, primeiramente definindo termos fundamentais como experiência aleatória, espaço amostral e evento. Depois, apresenta-se o valor da probabilidade de ocorrer um evento, obtido pela razão entre o número de resultados favoráveis pelo número dos possíveis. Finalmente, essa fórmula é aplicada para determinar a probabilidade de se obter um número par ao se jogar um dado.

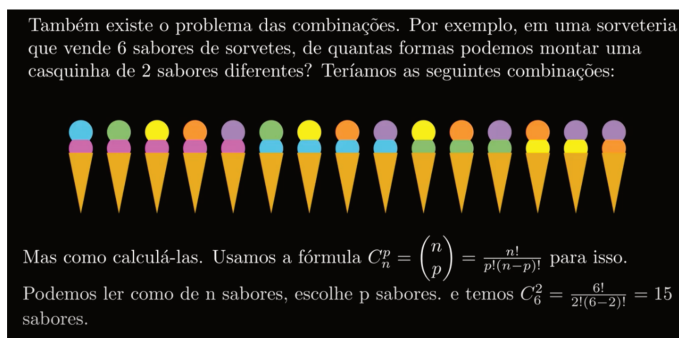


Figura 9: Cena da animação sobre combinatória e probabilidade.

4. Considerações finais

Para desenvolver animações com o **Manim** é necessário conhecer a programação em **Python**, pelo menos num nível básico. Por essa razão, desenvolvemos o manual para que iniciantes em **Python** possam primeiramente aprender a linguagem **Python** para depois então aprender a usar o **Manim**. O fato de o **Python** ser muito popular e que já vem sendo oferecido em escolas de Ensino Médio [5, 18], inclusive com materiais disponíveis para esse público[3], faz com que esperemos que o número de animações usando o **Manim** cresça, e que essas animações possam se tornar facilitadoras do aprendizado de Matemática e presentes no dia a dia de professores e alunos.

As animações produzidas têm o seu papel em complementar as aulas de Matemática. Por outro lado, o próprio processo de criação das animações também pode ser considerado uma atividade que complementa o ensino de Matemática. Esse processo requer, além do estudo do tema, criatividade para elaborar um roteiro que seja o mais didático possível. Além disso, durante a implementação do código da animação, conceitos da própria Matemática são usados para realizar as operações previstas no roteiro, expondo assim o animador a aplicações diretas da Matemática.

Agradecimentos

Agradecemos o apoio financeiro do Conselho Nacional de Desenvolvimento Científico e Tecnológico por meio da bolsa de iniciação tecnológica concedida a Eric S. S. Kishimoto (0220036212472856) para o desenvolvimento do manual e das animações. Agradecemos também ao Prof. José Carlos Magossi (FT/UNICAMP) pela leitura crítica do manuscrito e pelas sugestões de melhoria do trabalho.

Referências

- [1] Agustini, E.; Lopes, E. M. C. “Variações de Parâmetros em Funções: proposta e experiência didática remota com o GeoGebra.” *Professor de Matemática Online* (PMO) – Sociedade Brasileira de Matemática, v. 9, n° 1, pp. 36–56, 2021.
- [2] Alencar, H.; Cândido L.; Garcia R.; Mathias, C. “O GeoGebra como ferramenta de apoio ao entendimento de demonstrações em Geometria”. *Professor de Matemática Online* (PMO) – Sociedade Brasileira de Matemática, v. 10, n° 4, pp. 482–501, 2022.

- [3] Bodin A.; “Python in High School: Algorithms and Mathematics”, *Independently published*, 266p, 1ª Edição, 2010.
- [4] Coluci, V. R. “Animações de conceitos da teoria de erros usando Manim/Python”, *Revista Brasileira de Ensino de Física*, v. 44, e20210239, 2022.
- [5] Costa, A. C. M.; Graça, R. J. S.; Mota, C. V. A.; Alice Franco, A.; Muniz, V. H. J.; Maia, L. L.; Liese, T. M.; “Python: Será que é possível numa Escola Pública de Ensino Médio?”, *Anais do XXIII Workshop de Informática na Escola (WIE 2017)*, pp. 255-264, 2017.
- [6] Diógenes, R. “Geometria diferencial de curvas planas com Geogebra.” *Professor de Matemática Online (PMO)* – Sociedade Brasileira de Matemática, v. 7, nº 2, pp. 226–233, 2019.
- [7] ExpLORa, Faculdade de Tecnologia Unicamp. 2022, Página Inicial. <<https://wordpress.ft.unicamp.br/explora/>>. Acesso em: 5 de out. de 2022.
- [8] Nietto S. G.; Martins, M. R.; Derivando a Matemática, Instituto de Matemática, Estatística e Computação Científica Unicamp. 2022, Página Inicial. <<http://www.ime.unicamp.br/~apmat/>>. Acesso em: 5 de out. de 2022.
- [9] Geogebra. Geogebra, c2021. Página inicial. Disponível em: <<https://www.geogebra.org/>>. Acesso em: 5 de out. de 2022.
- [10] Goodstein, D. L.; Olenick, R. P. “Making The Mechanical Universe.” *American Journal of Physics*, v. 56, nº 9, pp. 779–785, 1988.
- [11] Lages Lima, E.; Carvalho, P. C. P.; Wagner E.; Morgado, A. C. *A Matemática do Ensino Médio*, Editora SBM, 237p, 11ª Edição, 2016.
- [12] Lima, J.; Freitas, L. “O uso do software GeoGebra para o estudo de problemas de otimização no ensino médio.” *Professor de Matemática Online (PMO)* – Sociedade Brasileira de Matemática, v. 6, nº 1, pp. 1–21, 2018.
- [13] Monzo, L. W. ; Gravina, M. A. “An Introduction to complex variable functions in high school: a possibility through the use of interactive animations.” *Bolema: Boletim de Educação Matemática*, v. 27, nº 46, pp. 645–661, 2013.
- [14] Moraes, C. F.; Amaral Rodriguez, B. D.; Poffal, C. A. “Relato de uma experiência: explorando conceitos de Geometria Analítica com o uso de animações no Geogebra”, *Revista Thema* v. 15 nº 1, pp. 333–349, 2018.
- [15] Notas sobre o país, Oecd, Pisa. Disponível em: <https://download.inep.gov.br/acoes_internacionais/pisa/resultados/2018/pisa_2018_brazil_prt.pdf/>. Acesso em: 5 de out. de 2022.
- [16] Oliveira, I. N. C.; Freitas, T. P. A. “O GeoGebra como recurso na promoção da interdisciplinaridade na educação básica: a geometria das bandeiras.” *Professor de Matemática Online (PMO)* – Sociedade Brasileira de Matemática, v. 8, nº 3, pp. 307–322, 2020.
- [17] Ortega, R.; Abbeg, T. “História, resolução numérica e GeoGebra no ensino de equações algébricas.” *Professor de Matemática Online (PMO)* – Sociedade Brasileira de Matemática, v. 4, nº 1, pp. 5–21, 2016.
- [18] Pesente, G. M.; “O ensino de matemática por meio da linguagem de programação Python.” Dissertação (Mestrado em Ensino de Ciência e Tecnologia) – Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2019.
- [19] Pinho, A.; Amâncio T.; “Prova do Pisa expõe década de estagnação no ensino no Brasil; China passa a liderar”. *Folha de S.Paulo*, São Paulo, 3 de

dez. de 2019. Disponível em: <https://www1.folha.uol.com.br/educacao/2019/12/prova-expoe-decada-de-estagnacao-no-ensino-no-brasil-china-passa-a-liderar.shtml>. Acesso em: 5 de out. de 2022.

- [20] Python Software Foundation, Welcome to Python.org, 2022, Página Inicial. Disponível em: <https://www.python.org/>. Acesso em: 5 de out. de 2022.
- [21] Ribeiro, R.; Gravina, M. “Disco de Poincaré: uma proposta para explorar geometria hiperbólica no GeoGebra.” *Professor de Matemática Online* (PMO) – Sociedade Brasileira de Matemática, v. 1, n° 1, pp. 53–66, 2013.
- [22] Rodrigues, G.; Oliveira, E. “O uso do GeoGebra no ensino de elipse nas aulas de matemática da Educação Básica.” *Professor de Matemática Online* (PMO) – Sociedade Brasileira de Matemática, v. 5, n° 2, pp. 11–30, 2017.
- [23] Taylor, M.; Pountney, D.; Malabar, I. “Animation as an aid for the teaching of mathematical concepts”, *Journal of Further and Higher Education*, v. 31, n° 3, pp. 249–261, 2007.

Eric Satoshi Suzuki Kishimoto
Faculdade de Tecnologia, Universidade Estadual de Campinas
e233974@dac.unicamp.br

Vitor Rafael Coluci
Faculdade de Tecnologia, Universidade Estadual de Campinas
coluci@unicamp.br

Recebido: 7/10/2022
Publicado: 15/02/2023