

## 14<sup>o</sup> Congresso de Inovação, Ciência e Tecnologia do IFSP - 2023

### LINGUAGEM PYTHON NO LABORATÓRIO DIDÁTICO DE FÍSICA: UMA ALTERNATIVA PARA ANÁLISE DE DADOS

ALISON JOSÉ PEREIRA NERI<sup>1</sup>, VÁGNER RICARDO DE ARAÚJO PEREIRA<sup>2</sup>

<sup>1</sup> Graduando em Eng. de Controle e Automação, Bolsista PIBFISP, IFSP, Câmpus Catanduva, alison.n@aluno.ifsp.edu.br

<sup>2</sup> Professor Doutor, IFSP, Câmpus Catanduva, vagner.pereira@ifsp.edu.br

Área de conhecimento (Tabela CNPq): 1.05.01.05-3 Metrologia, Técnicas Gerais de Laboratório, Sistema de Instrumentação.

**RESUMO:** A análise de dados no laboratório didático de física, à luz da teoria, é fundamental para a compreensão do método científico em pesquisa experimental. Propõe-se avaliar a linguagem de programação *Python* como alternativa gratuita, em relação aos softwares *Excel* (2023) e *OriginPro* (2019). Para isso, foram comparados os parâmetros fixos de ajuste de curvas desses softwares com aqueles obtidos em um *script* desenvolvido em *Python*, disponibilizado no *GitHub*. O experimento abordou o movimento retilíneo com aceleração constante usando um trilho de ar. Os resultados mostraram-se iguais quando comparado ao *OriginPro* (2019) e o *Excel* (2023) com aproximação de quatro casas decimais. Assim, o *Python* mostrou ser uma sólida alternativa para análise de dados, com equivalência analítica, flexibilidade e facilidade de obtenção. A abordagem interdisciplinar que une *Python* e física permite aos alunos integrarem teoria, experimentação e programação, ampliando o acesso e desenvolvendo habilidades valiosas tanto em termos de aprendizagem quanto para o mercado de trabalho.

**PALAVRAS-CHAVE:** aceleração constante; ajuste de curva; linguagem de programação; movimento retilíneo; prática de laboratório; trilho de ar.

### PYTHON LANGUAGE IN PHYSICS DIDACTIC LABORATORY: AN ALTERNATIVE FOR DATA ANALYSIS

**ABSTRACT:** Data analysis in the didactic physics laboratory, in light of theory, is fundamental for the understanding of the scientific method in experimental research. It is proposed to evaluate the Python programming language as a free alternative in comparison to the Excel (2023) software and OriginPro (2019). To achieve this, the fixed curve fitting parameters of these software applications were compared with those obtained from a script developed in Python, made available on the GitHub. The experiment addressed rectilinear motion with constant acceleration using an air track. The results demonstrated similarity when compared with OriginPro (2019) and with Excel (2023), with an approximation of four decimal places. Thus, Python has proven to be a robust alternative for data analysis, offering analytical equivalence, flexibility, and ease of access. The interdisciplinary approach that combines Python and physics enables students to integrate theory, experimentation, and programming, expanding access and developing valuable skills both in terms of learning and for the job market.

**KEYWORDS:** air track; constant acceleration; curve fitting; laboratory practice; programming language; rectilinear motion.

## INTRODUÇÃO

A atividade prática no laboratório didático de física permite aos estudantes desenvolver suas capacidades de raciocínio lógico, tornando a compreensão da disciplina mais acessível quando relacionada a situações concretas do dia a dia (Silva et al., 2020). O desenvolvimento do raciocínio lógico-matemático envolve a análise de dados obtidos empiricamente.

A utilização de *softwares* traz benefícios ao simplificar e agilizar tarefas em diversas situações (Alfonsi, 2005; Santos et al., 2005). Entretanto, vários deles são pagos, como os *softwares Excel* (2023) e *OriginPro* (2019), o que pode resultar em subutilização devido à falta de recursos financeiros nas instituições de ensino ou, em alguns casos, ao uso não autorizado. A ausência desses programas de análise de dados pode prejudicar a adequada preparação dos estudantes nos laboratórios, pois eles não conseguirão interpretar e estudar os fenômenos físicos adequadamente.

Diante desse desafio, a linguagem *Python* emerge como uma solução gratuita para a análise de dados, oferecendo uma diversidade de plataformas de programação. Segundo Souza Filho et al. (2009), o *Python* é uma ferramenta eficaz para superar obstáculos no laboratório de física. Além disso, conforme apontado por Rolim (2021), a utilização de linguagens de programação em disciplinas de ciências exatas pode incentivar os estudantes a se envolverem na resolução de problemas por meio da programação.

Nesse contexto, o objetivo desta pesquisa é verificar se a linguagem *Python* é uma alternativa confiável para análise de dados. Isso será feito por meio da criação de um *script*, com a verificação dos resultados obtidos em um experimento de movimento retilíneo com aceleração constante. Além disso, será realizada uma comparação com o *Excel* (2023) e o *OriginPro* (2019), com o intuito de determinar se o *Python* pode eficazmente substituir esses *softwares* pagos comumente utilizados.

## MATERIAIS E MÉTODOS

Com o auxílio de um kit experimental de trilho de ar linear da marca *Azeheb*, foi possível realizar a coleta de dados experimentais sobre movimento linear com aceleração constante para validar a análise do *script*. Os componentes usados no experimento podem ser visualizados na figura 1 a seguir.

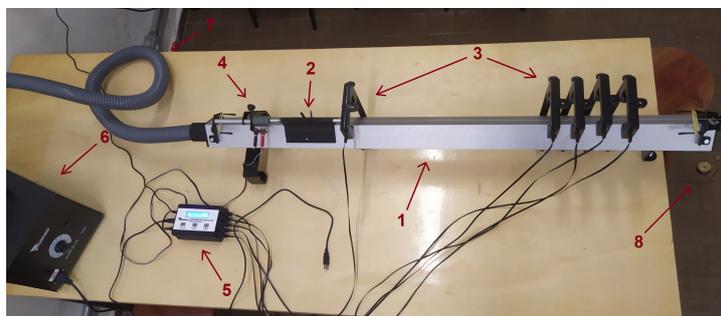


Figura 1: Conjunto experimental para movimento retilíneo com aceleração constante.

Componentes e suas respectivas numerações:

- |              |                            |                    |                                       |
|--------------|----------------------------|--------------------|---------------------------------------|
| 1. Trilho;   | 3. Sensores fotoelétricos; | 5. Cronômetro;     | 7. Tubulação de ar;                   |
| 2. Carrinho; | 4. Eletroímã;              | 6. Soprador de ar; | 8. Massa presa ao carrinho por linha. |

Inicialmente, o carrinho foi fixado no eletroímã. Os sensores foram posicionados e suas distâncias relativas foram medidas com uma trena de resolução de 0,001 m (1 mm). Em seguida, o soprador de ar foi ativado, criando um fluxo de ar no trilho. Esse fluxo de ar forma uma camada que reduz significativamente o atrito. Posteriormente, usando um cronômetro digital, o eletroímã foi desativado, permitindo que o carrinho começasse a se mover com aceleração constante, uma vez que o carrinho foi puxado por uma linha que tinha uma massa em sua extremidade.

À medida que o carrinho percorre o trilho de ar, os sensores fotoelétricos registram o tempo decorrido desde o desligamento do eletroímã. A resolução do tempo medido é de 0,000001 s (1  $\mu$ s).

Devido à limitação do cronômetro, que possui somente cinco entradas para sensores fotoelétricos, foi adotada a estratégia de manter fixo o primeiro sensor que o carrinho atravessa e deslocar os outros quatro sensores, de modo a coletar um volume maior de dados para gerar um gráfico mais preciso. Esse processo foi repetido diversas vezes até que uma quantidade significativa de dados fosse obtida. A cada nova execução do movimento do carrinho, foi assegurada a coerência dos resultados para garantir a confiabilidade das medições.

A linguagem *Python* foi empregada na criação de um *script* que opera com base nos dados obtidos no experimento descrito acima, cujo o usuário insere os dados de posição dos sensores e o tempo entre a passagem do objeto de um sensor para o outro, a parte do *script* responsável pela requisição de dados é as linhas 8, 10 e 11 que podem ser visualizadas na figura 2. Isso resulta na geração de um gráfico contendo os parâmetros fixos da curva correspondentes ao movimento retilíneo com aceleração constante. Conforme descrito por Halliday, Resnick e Walker (2016), um movimento com essas características é expresso pela equação 1.

$$x(t) = x_0 + v_0 \cdot t + \frac{a \cdot t^2}{2} \quad (1)$$

em que,

$x(t)$  – posição do objeto em relação ao tempo, m;

$x_0$  - posição inicial do objeto, m;

$v_0$  – velocidade inicial do objeto, m/s;

$t$  - tempo decorrido a partir da posição inicial, s;

$a$  - aceleração do objeto, m/s<sup>2</sup>.

O *script* efetua a impressão do gráfico e a realização dos ajustes de curvas por meio de três bibliotecas instaladas, que são incluídas no código nas linhas 1, 2 e 3 que podem ser visualizadas na figura 2, as bibliotecas são: a *matplotlib.pyplot*<sup>1</sup> para criação de gráficos, a *scipy.optimize*<sup>2</sup> para efetuar o ajuste de curva e obter os parâmetros fixos da curva, e a biblioteca *numpy*<sup>3</sup>, que engloba diversas funcionalidades matemáticas. Além do uso dessas bibliotecas, foi implementada uma função nas linhas 12 e 13 (figura 2) que corresponde à equação 1, sem a divisão por dois no último termo da equação, estabelecendo uma conexão entre os valores experimentais e o modelo teórico.

Pontos de importante destaque do *script* incluem a plotagem dos dados no gráfico, onde ocorre a passagem dos dados informados pelo usuário para os eixos x e y (linhas 14 e 15). Posteriormente, esses dados são plotados no gráfico, onde são definidos visualmente os pontos e a legenda (linha 19). Além disto, tem-se o cálculo dos parâmetros fixos pela função mencionada anteriormente (linhas 20 e 21),

---

<sup>1</sup><<https://matplotlib.org>>

<sup>2</sup><<https://scipy.org>>

<sup>3</sup><<https://numpy.org>>

bem como a plotagem deles no gráfico (linha 22). As linhas mencionadas podem ser visualizadas na Figura 2. O *script* completo em linguagem *Python* está disponível no no *website* *GitHub*<sup>4</sup>.

```

1 import matplotlib.pyplot as plt
2 from scipy.optimize import curve_fit
3 import numpy as np

8 t = np.array([float(input('Informe quantos segundos deu no ' + str(i+1) + 'º sensor (em
segundos): ')) for i in range(n)])

10 for i in range(1, n):
11     x.append(float(input('Informe a distância do ' + str(i) + 'º ao ' + str(i+1) + 'º sensor
(em metros): ')))

14 xData = np.array(t)
15 yData = np.array(x)

12 def y(t, C, v0, x0):
13     return (C*t**2) + v0*t + x0

19 plt.plot(xData, yData, 'bo', label='Dados')
20 popt, pcov = curve_fit(y, xData, yData)
21 xFit = np.arange(0.0, t[n-1], 0.000001)
22 plt.plot(xFit, y(xFit, *popt), 'r', label=f'Parâmetros fixos de ajuste:\n C={popt
[0]:.5} m/s**2, v0={popt[1]:.5} m/s, x0={popt[2]:.5e} m\nEquação: x = C*t**2 + v0
*t + x0')

```

Figura 2: Linhas principais do *script*.

Para avaliar a qualidade do gráfico e dos parâmetros fixos de ajuste de curva obtidos pelo *script*, foram criados outros dois gráficos usando os mesmos dados, porém com dois programas distintos: *Excel* (2023) e *OriginPro* (2019). Esses *softwares* são frequentemente empregados tanto no âmbito educacional quanto empresarial para a análise de dados.

## RESULTADOS E DISCUSSÃO

A Tabela 1 apresenta os resultados do experimento de movimento retilíneo com aceleração constante. A primeira coluna da esquerda apresenta os dados de tempo obtidos pelo sensor fotoelétrico acoplado no trilho de ar. Na segunda coluna, realizou-se a subtração do primeiro dado de tempo, uma vez que o primeiro sensor fotoelétrico é considerado o ponto de partida do movimento. A terceira coluna contém os dados de posição dos sensores fotoelétricos em relação ao primeiro sensor.

Tabela 1: Dados experimentais de tempo absoluto e relativo, além de dados de posição relativa.

t (s)	$\Delta t$ (s)	x (m)
0,300293	0,000000	0,000
0,358864	0,058571	0,050
0,408784	0,108491	0,100
0,454299	0,154006	0,150
0,495479	0,195186	0,200
0,533066	0,232773	0,250
0,569130	0,268837	0,300
0,602781	0,302488	0,350
0,634345	0,334052	0,400
0,664218	0,363925	0,450
0,691898	0,391605	0,500
0,720067	0,419774	0,550
0,746927	0,446634	0,600

<sup>4</sup><<https://github.com/Imp0SS1ble/CONICT/blob/c0b7021ee56360aedd3ef787ecb1a01be14ec949/Script.py>>

Após coletar os dados experimentais, os dados da segunda e terceira colunas foram compilados e inseridos no *script*. Ao realizar essa ação, o gráfico (Figura 3) foi gerado, com pontos azuis que representam os dados experimentais, além de uma curva vermelha que ilustra a linha de tendência dos dados. Os parâmetros fixos desta curva de tendência estão presentes na legenda e correspondem a uma curva de movimento retilíneo com aceleração constante, de acordo com a teoria.

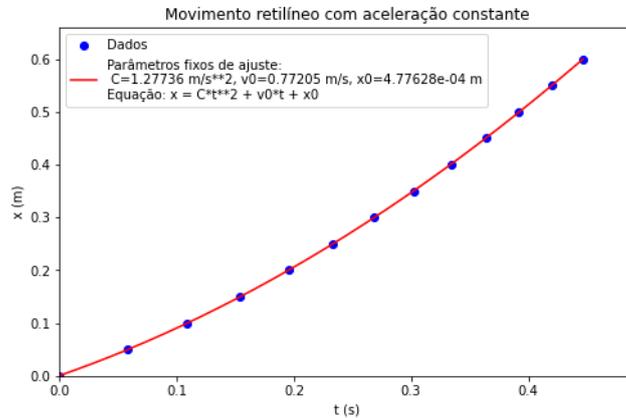


Figura 3: Gráfico obtido no *script* a partir dos dados experimentais de movimento retilíneo com aceleração constante, contendo também em sua legenda os parâmetros de ajuste.

A figura 4 apresenta os gráficos criados nos softwares Excel (2023) e OriginPro (2019) utilizando os mesmos dados.

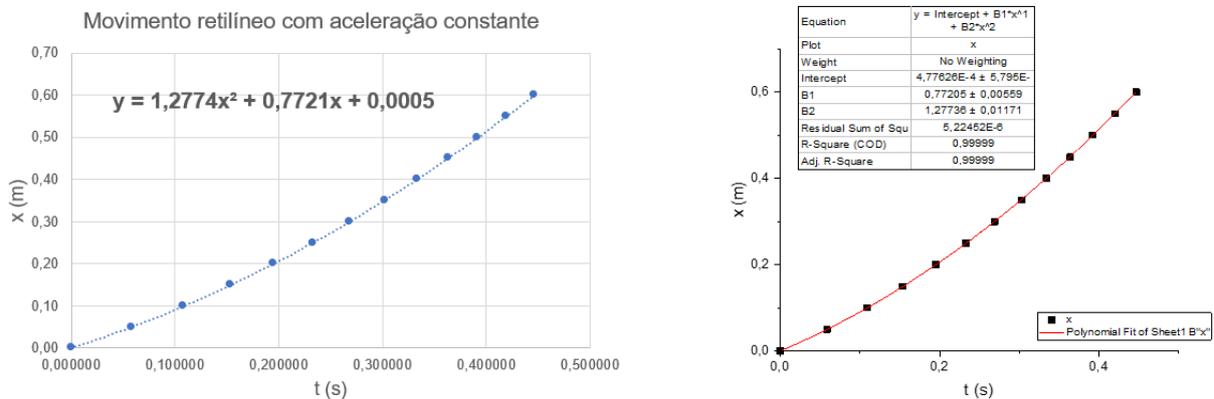


Figura 4: Gráfico a esquerda obtido no *Excel* (2023) e gráfico a direita obtido no *OriginPro* (2019).

Os parâmetros fixos  $1,27736$ ,  $0,77205$  e  $4,77628 \cdot 10^{-4}$ . Representam respectivamente, metade do valor da aceleração do carrinho ( $m/s^2$ ), velocidade inicial do carrinho ( $m/s$ ) e posição inicial do carrinho ( $m$ ).

Os resultados obtidos para os parâmetros de ajuste de curva, utilizando o mesmo modelo de equação, entre o *script* e o *OriginPro* (2019) os valores foram idênticos. Em relação entre o *script* e o *Excel* (2023) devido à limitação do *software* em exibir apenas quatro casas decimais, houve arredondamentos na quarta casa decimal nos parâmetros fixos da curva de ajuste.

## CONCLUSÕES

Diante dos dados obtidos é possível concluir que a linguagem *Python* se destaca como uma alternativa notável em relação aos *softwares* de análise de dados. Isso não apenas se deve à sua equivalência

na análise, mas também à sua versatilidade e alto nível de customização. Além disso, o uso do *Python* oferece aos estudantes uma abordagem interdisciplinar, na qual eles podem aprender os conceitos teóricos da física, realizar experimentos práticos ou simulados e, por fim, programar um *script* para analisar seus dados experimentais. Essa abordagem proporciona aos estudantes uma polivalência valiosa e altamente procurada pelo mercado de trabalho atual, conforme destacado por Benevides Filho (1999) em sua pesquisa de mestrado.

Além disso, vale ressaltar que os *softwares* utilizados para a comparação de ajuste de curva são pagos e representam as ferramentas mais utilizadas para análise de dados. Em contraste, o *Python* é uma opção gratuita e oferece uma ampla variedade de compiladores disponíveis sem custo. Isso possibilita um acesso a uma ferramenta de análise de dados para todos os estudantes.

## CONTRIBUIÇÕES DOS AUTORES

Alison José Pereira Neri: Coleta e análise de dados, desenvolvimento de *script*, disponibilização de *softwares*, *design*, validação e apresentação de dados e redação do manuscrito. Vágner Ricardo de Araújo Pereira: Concepção do projeto, orientação, obtenção de financiamento, disponibilização da estrutura de laboratório e revisão do manuscrito.

## AGRADECIMENTOS

Agradecemos ao PIBIFSP pela bolsa de iniciação científica.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ALFONSI, B. Open source in the classroom. **Distributed Systems Online, IEEE**. v. 6, n. 6, p. 1-4, 2005.
- BENEVIDES FILHO, S. A. **A Polivalência como ferramenta para a produtividade**. Florianópolis: Universidade Federal de Santa Catarina, 1999.
- Excel**. Versão 2307. [Planilha eletrônica]. Microsoft Corporation, 2023.
- HALLIDAY, D.; RESNICK, R.; WALKER, J. **Fundamentos de Física: Mecânica**. v. 1, 10. ed. Rio de Janeiro: LTC, 2016.
- OriginPro**. Versão 2019b. [Construção e análise de gráficos]. OriginLab Corporation, 2019.
- ROLIM, E. L. S. S. **Uma unidade de ensino potencialmente significativa intermediada pela linguagem Python para o ensino de álgebra**, Unipampa, [TCC], 2021. Disponível em: <<https://dspace.unipampa.edu.br/handle/rii/5804>>. Acesso em 29 jun.2023.
- SANTOS, G. et al. Knowledge management in a software development environment to support software processes deployment. **Professional Knowledge Management Lecture Notes in Artificial Intelligence** 3782. Berlim p.111-120, 2005.
- SILVA, F. R. O.; MAIA, L. S. P.; CAVALCANTE, D. N. S. **Hands-on no aprendizado de ondulatória: Estudo de caso com uma plataforma robótica**. Braz. J. of Develop., Curitiba, v. 6, n. 3, p. 10526-10538, 2020.
- SOUZA FILHO, C. A. et al. Uso do Python como laboratório virtual na Física. IX Jornada de Ensino, Pesquisa e Extensão - **IX JEPEX**. Recife, PE – 19 a 23 de outubro de 2009. Disponível em: <<http://www.eventosufrpe.com.br/jepex2009/cd/resumos/R0966-1.pdf>>. pdf. Acesso em: 29 jun. 2023.