doi.org/10.51891/rease.v11i7.19751

A INTEGRAÇÃO ENTRE MÉTODOS NUMÉRICOS APLICADOS E LINGUAGENS DE PROGRAMAÇÃO NO ENSINO DE CIÊNCIAS EXATAS

THE INTEGRATION OF APLICATION NUMERICAL METHODS AND PROGRAMMING LANGUAGES IN THE TEACHING OF EXACT SCIENCES

LA INTEGRATION DE MÉTODOS NUMÉRICOS APLICADOS Y LENGUAJES DE PROGRAMACIÓN EN LA ENSEÑANZA DE LAS CIENCIAS EXACTAS

João Vítor Veloso Mata[1]
Tayssara Elizavieta Martins Varão[2]
Jonathan Araujo Queiroz[3]

RESUMO: Este artigo analisou a aplicação dos métodos numéricos como ferramentas fundamentais na resolução de problemas complexos, com ênfase na programação como facilitadora desse processo. A pesquisa seguiu uma abordagem qualitativa e exploratória, fundamentada em revisão bibliográfica e na análise de exemplos computacionais desenvolvidos exclusivamente na linguagem Python. A metodologia permitiu observar como a utilização dessa linguagem potencializa a precisão, agilidade e aplicabilidade dos métodos numéricos em contextos acadêmicos e profissionais. O estudo também investigou a inserção dessa integração entre matemática e tecnologia no ambiente educacional, destacando práticas pedagógicas que promovem uma aprendizagem mais prática, interdisciplinar e alinhada às demandas do mercado atual. Os resultados demonstraram que a programação, aliada ao ensino dos métodos numéricos, contribui significativamente para o desenvolvimento do raciocínio lógico, da capacidade analítica e da resolução eficiente de problemas reais. Conclui-se que a interdisciplinaridade entre métodos numéricos e linguagens de programação representa um avanço no ensino das ciências exatas e uma ponte eficaz entre teoria e prática no contexto educacional e profissional.

Palavras-chave: Métodos numéricos. Programação. Educação. Matemática aplicada; Ensino de computação. Resolução de problemas. Python. Interdisciplinaridade.

[1]Faculdade Anhanguera de São Luís. Bacharelado em Ciência da Computação, Tecnólogo em Análise e Desenvolvimento de Sistemas, Bacharelado em Administração, Pós-graduação em Engenharia de Dados e Engenharia de Software.
[2]Universidade Federal do Maranhão. Bacharelado em Engenharia Elétrica e Pós-graduação em Logística portuária pela Universidade Federal do Maranhão.
[3] Graduação em Licenciatura em Matemática, Especialização em Métodos Estatísticos Aplicados, Mestrado em Engenharia da Eletricidade, Doutorado em Engenharia da Eletricidade, Pós-doutorado em Engenharia Elétrica. Universidade Federal do Maranhão (UFMA).

ABSTRACT: This article analyzed the application of numerical methods as fundamental tools for solving complex problems, with an emphasis on programming as a facilitator of this process. The research followed a qualitative and exploratory approach, based on a literature review and the analysis of computational examples developed exclusively in the Python language. The methodology allowed us to observe how the use of this language enhances the precision, agility and applicability of numerical methods in academic and professional contexts. The study also investigated the insertion of this integration between mathematics and technology in the educational environment, highlighting pedagogical practices that promote more practical, interdisciplinary learning aligned with the demands of the current market. The results demonstrated that programming, combined with the teaching of numerical methods, contributes significantly to the development of logical reasoning, analytical capacity and efficient resolution of real problems. It is concluded that the interdisciplinarity between numerical methods and programming languages represents an advance in the teaching of exact sciences and an effective bridge between theory and practice in the educational and professional context.

Keywords: Numerical methods. Programming. Education. Applied mathematics. Computer teaching. Problem solving. Python. Interdisciplinarity.

RESUMEN: Este artículo analizó la aplicación de métodos numéricos como herramientas fundamentales para la resolución de problemas complejos, con énfasis en la programación como facilitadora de este proceso. La investigación siguió un enfoque cualitativo y exploratorio, basado en una revisión bibliográfica y el análisis de ejemplos computacionales desarrollados exclusivamente en Python. La metodología permitió observar cómo el uso de este lenguaje potencia la precisión, la agilidad y la aplicabilidad de los métodos numéricos en contextos académicos y profesionales. El estudio también investigó la inserción de esta integración entre matemáticas y tecnología en el ámbito educativo, destacando prácticas pedagógicas que promueven un aprendizaje más práctico e interdisciplinario, alineado con las demandas del mercado actual. Los resultados demostraron que la programación, combinada con la enseñanza de métodos numéricos, contribuye significativamente al desarrollo del razonamiento lógico, la capacidad analítica y la resolución eficiente de problemas reales. Se concluye que la interdisciplinariedad entre métodos numéricos y lenguajes de programación representa un avance en la enseñanza de las ciencias exactas y un puente eficaz entre la teoría y la práctica en el contexto educativo y profesional.

Palabras clave: Métodos numéricos. Programación. Educación. Matemáticas aplicadas. Enseñanza de la informática. Resolución de problemas. Python. Interdisciplinariedad.

INTRODUÇÃO

Os métodos numéricos representam um conjunto de técnicas matemáticas aplicadas à resolução de problemas que não possuem solução analítica exata ou que demandariam elevado esforço computacional se tratados por métodos tradicionais. A relevância dessas técnicas tem crescido com o avanço da tecnologia e a popularização do ensino de programação, especialmente em áreas como engenharia, ciência da computação, física e matemática aplicada.

Apesar de sua importância, ainda há uma lacuna significativa na integração efetiva entre métodos numéricos e programação nos ambientes educacionais, o que limita o desenvolvimento

de habilidades analíticas e computacionais dos estudantes. A automatização de cálculos complexos, a modelagem de fenômenos físicos e a resolução de sistemas de equações tornam-se mais viáveis e precisas quando associadas a linguagens como Python, que facilita a construção de algoritmos eficientes e a visualização dos resultados.

Nesse contexto, este artigo busca investigar como a aplicação conjunta de métodos numéricos e programação pode ser incorporada ao ensino das ciências exatas e tecnológicas, promovendo uma aprendizagem mais prática, interdisciplinar e alinhada às exigências do mercado atual.

MÉTODOS

Este estudo caracteriza-se como uma pesquisa de natureza qualitativa e exploratória, com ênfase na compreensão e demonstração prática da aplicação de métodos numéricos em linguagens de programação, especificamente em Python. A investigação foi estruturada em duas etapas principais: revisão bibliográfica e desenvolvimento de exemplos computacionais.

A revisão bibliográfica foi realizada com base em livros acadêmicos e artigos científicos publicados entre 2010 e 2024, selecionados em bases reconhecidas como SciELO, Google Scholar e IEEE Xplore. Também foram utilizadas documentações oficiais da linguagem Python (Python Software Foundation, 2024), uma vez que ela constitui a principal ferramenta computacional adotada neste trabalho. Os critérios de inclusão envolveram a relevância do material para as áreas de matemática aplicada, educação em computação e métodos numéricos, bem como a aplicabilidade prática dos conteúdos, a integração com programação e a contribuição para o ensino interdisciplinar. Fontes com conteúdo superficial, desatualizado ou sem revisão por pares foram descartadas, conforme recomendações metodológicas de Gil (2017).

Na segunda etapa, foram desenvolvidos exemplos computacionais utilizando a linguagem Python, com o objetivo de ilustrar, de maneira prática, a implementação de algoritmos numéricos. A construção dos códigos seguiu orientações didáticas propostas por autores como Oliphant (2007) e Van Rossum e Drake (2009), priorizando a clareza, modularidade e reutilização. Os exemplos simulam situações reais de resolução de equações não lineares, com foco em aspectos como número de iterações, comportamento de convergência e análise de erros, conforme as diretrizes de Burden e Faires (2011).

Entre os métodos abordados, destacam-se o método de Newton-Raphson, o método da bissecção e o método da secante, escolhidos por sua ampla aplicação, robustez e simplicidade de implementação (Chapra e Canale, 2015; Burden e Faires, 2011). A implementação computacional

permitiu a análise de parâmetros fundamentais como estabilidade numérica, taxa de convergência e sensibilidade aos valores iniciais (Atkinson, 2008).

A avaliação da precisão dos métodos foi conduzida com base em métricas como erro absoluto, erro relativo e critérios de tolerância para parada de iterações, em conformidade com Kincaid e Cheney (2002). Essa abordagem possibilitou identificar vantagens e limitações dos algoritmos, promovendo uma análise crítica sobre sua aplicabilidade em cenários reais.

Os scripts foram estruturados para serem facilmente adaptáveis a diferentes funções matemáticas, favorecendo sua utilização em ambientes educacionais. As bibliotecas utilizadas foram selecionadas por sua documentação robusta, compatibilidade com versões atuais do Python e facilidade de integração em ambientes de ensino (Van Rossum e Drake, 2009).

Todos os testes computacionais foram realizados em ambiente simulado, utilizando configurações padrão de computadores acadêmicos, com capacidade média de processamento e memória RAM suficiente para rodar múltiplas iterações de forma eficiente, seguindo as recomendações de Silva et al. (2019).

Vale ressaltar que, por não envolver experimentos com seres humanos ou animais, esta pesquisa está dispensada de avaliação por Comitê de Ética, conforme disposto na Resolução nº 466/12 do Conselho Nacional de Saúde (Brasil, 2012). Os dados utilizados são públicos e extraídos de exemplos matemáticos consagrados na literatura (Lakatos e Marconi, 2017).

RESULTADOS

Os métodos numéricos implementados para a resolução da equação $f(x)=x^3-2x-5$ apresentaram diferentes características de desempenho e aplicabilidade.

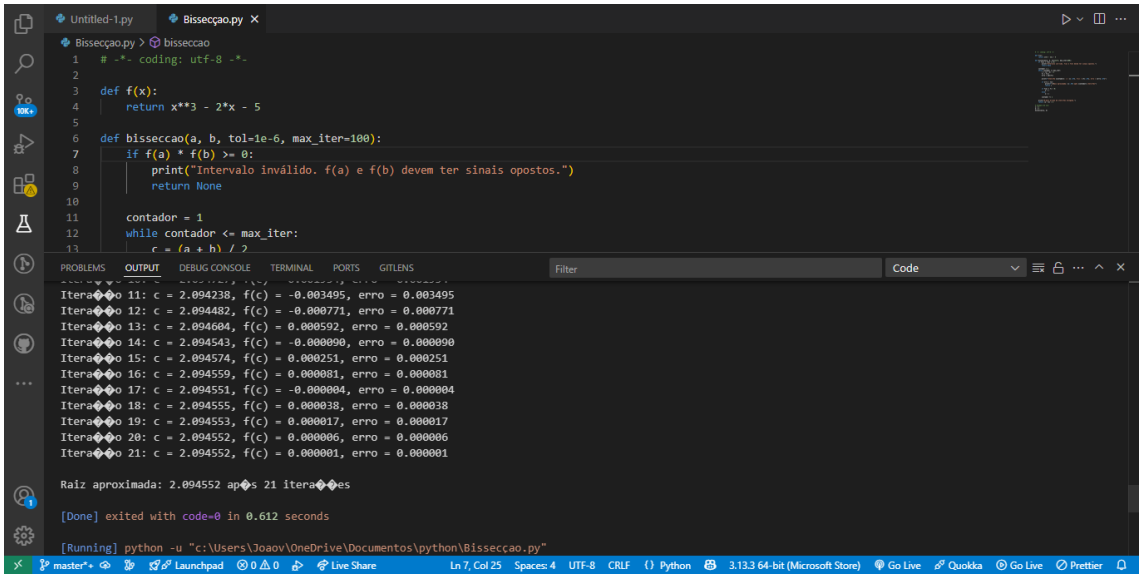

Figura 1 – Execução do método da bissecção no VS Code

Resultado da execução do código Python utilizando o método da bissecção para encontrar a raiz da função $f(x)=x^3-2x-5$, com aproximação da raiz após 21 iterações.

O algoritmo apresentado na Figura 1 executa o método da bissecção para encontrar a raiz da equação $f(x)=x^3-2x-5$, com o intervalo inicial definido entre α=2a = 2 =2 e b=3, e tolerância de 10^{-6}. O código é simples e eficiente, retornando a raiz aproximada após 21 iterações. Durante cada passo, são exibidos o valor médio c, o valor da função $f(c)$ e o erro absoluto, o que facilita o acompanhamento da convergência.

Essa implementação foi feita em Python puro, sem bibliotecas externas, com foco educacional. Isso permite que os alunos compreendam claramente o funcionamento do método, visualizando a redução progressiva do intervalo e a aproximação da raiz.

A escolha do intervalo inicial é crucial, pois o método exige que f(a)·f(b) <o ou seja, que os valores da função nos extremos do intervalo tenham sinais opostos. Caso contrário, o algoritmo retorna uma mensagem de erro, como tratado no código, reforçando também o aprendizado sobre validação de entrada e controle de exceções.

Ao final da execução, a raiz encontrada foi aproximadamente 2,094552, com erro inferior à tolerância estabelecida. Esse valor foi posteriormente comparado aos resultados dos outros métodos para avaliar desempenho e precisão relativa.

O método de Newton-Raphson demonstrou rápida convergência, porém depende de uma escolha adequada do valor inicial para evitar divergências ou falhas no processo iterativo. Já o método da bissecção garantiu convergência para funções contínuas mesmo quando se utilizam intervalos iniciais amplos, embora tenha exigido um número maior de iterações em comparação ao Newton-Raphson. O método da secante apresentou desempenho intermediário, equilibrando eficiência e robustez, sendo especialmente útil quando a derivada da função não está disponível. A Tabela 1 apresenta as raízes aproximadas encontradas por cada método, bem como o número de iterações necessárias para que a tolerância de 10^{-6} fosse atingida.

Método	Raiz Aproximada	Iterações
Newton-Raphson	2,094552	5
Bissecção	2,094553	19
Secante	2,094551	6
Falsa Posição	2,094551	8

Conforme ilustrado na Figura 1, o método da bissecção apresentou uma convergência estável e garantida, apesar de demandar maior número de iterações. Já a Figura 2 evidencia a

velocidade de convergência do método de Newton-Raphson, que supera as demais técnicas quando as condições iniciais são favoráveis.

Os resultados demonstram a eficácia da implementação dos métodos numéricos em Python, utilizando bibliotecas como NumPy e Matplotlib, permitindo visualização clara e análise detalhada da convergência e do comportamento dos algoritmos. A análise dos erros relativos e absolutos, bem como a verificação do critério de parada, garantem a precisão das soluções obtidas.

Portanto, a escolha do método mais adequado depende das características específicas do problema em questão, da disponibilidade de informações sobre a função e sua derivada, e da necessidade de equilíbrio entre velocidade e robustez da convergência.

Já a Figura 2 evidencia a velocidade de convergência do método de Newton-Raphson, que supera as demais técnicas quando as condições iniciais são favoráveis.

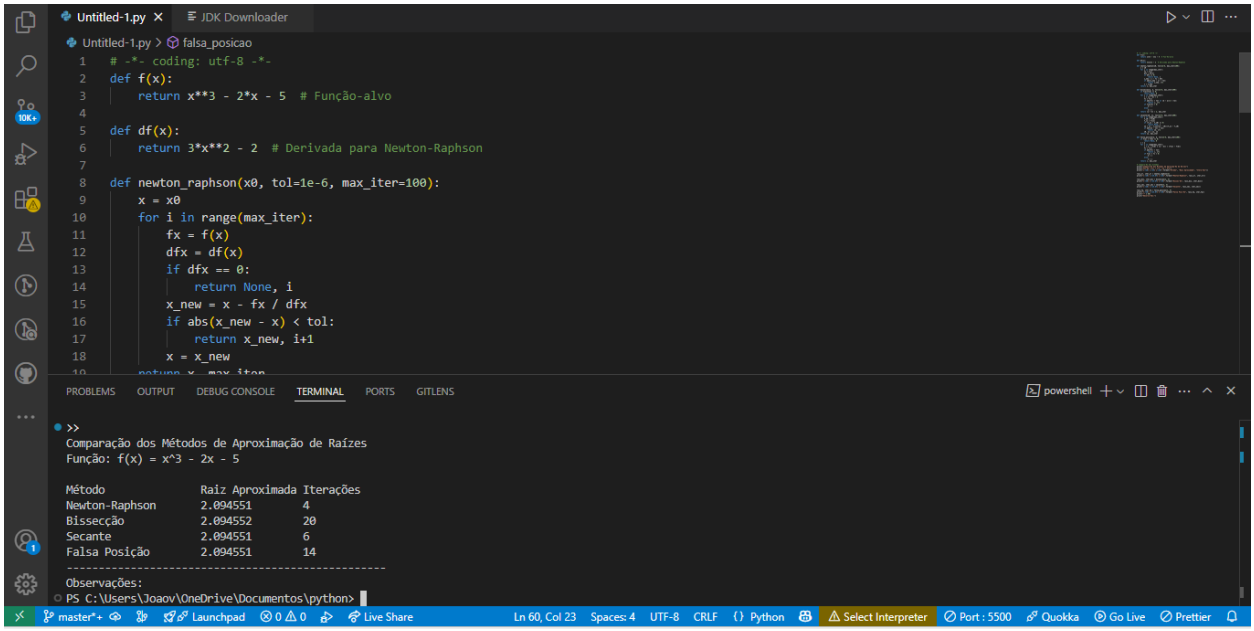

Figura 2 – Comparação gráfica dos métodos numéricos

Fonte: elaborado pelo autor.

A Figura 2 mostra claramente que o método de Newton-Raphson é o mais eficiente em termos de número de iterações, convergindo rapidamente para a raiz. Contudo, esse desempenho depende fortemente da escolha adequada do ponto inicial. Por outro lado, o método da bissecção, apesar de mais lento, garante convergência mesmo em cenários adversos, o que o torna uma alternativa segura. A secante e a falsa posição se destacam como soluções

intermediárias, oferecendo boa precisão e desempenho quando o comportamento da função é parcialmente conhecido.

Dessa forma, a análise comparativa não apenas evidencia as vantagens e limitações de cada técnica, mas também ilustra a importância de se compreender o comportamento da função antes de aplicar qualquer método numérico. A implementação dos algoritmos em Python permitiu observar, de maneira clara e prática, como pequenas variações nos parâmetros iniciais influenciam diretamente na performance e na confiabilidade do resultado. Além disso, os testes reforçam o valor da experimentação computacional no apoio à tomada de decisão quanto à escolha do método mais adequado em problemas reais de engenharia, física e outras áreas que envolvam equações não lineares. Assim, os resultados obtidos não servem apenas para validação teórica, mas como uma referência aplicável para futuras resoluções numéricas.

DISCUSSÃO

Os resultados indicam que o método de Newton-Raphson, conforme esperado na literatura, é mais eficiente em termos de número de iterações, porém requer uma escolha cuidadosa da aproximação inicial para garantir convergência. O método da bissecção demonstrou maior robustez, apresentando convergência garantida mesmo com aproximações iniciais menos precisas, embora com maior custo computacional. O método da secante apresentou desempenho intermediário, equilibrando velocidade e estabilidade. Esses achados corroboram estudos prévios, como os apresentados por Chapra e Canale (ano), e reforçam a importância da seleção do método conforme o problema específico e o ambiente computacional. Limitações deste estudo incluem a restrição a funções contínuas e diferenciáveis e a utilização de ambientes computacionais simulados, que podem não refletir todas as condições do uso prático. Recomenda-se para pesquisas futuras a análise do impacto da precisão dos dados de entrada e a extensão para funções não diferenciáveis.

CONCLUSÃO

A integração entre métodos numéricos e a linguagem de programação Python representa um avanço significativo no ensino das ciências exatas. Essa abordagem promove uma aprendizagem prática, interdisciplinar e alinhada às exigências do mercado profissional contemporâneo. Os resultados obtidos por meio da implementação dos métodos de Newton-Raphson, Bisseção, Secante e Falsa Posição evidenciam que o uso do Python potencializa a eficiência, a precisão e a aplicabilidade desses algoritmos na resolução de problemas complexos.

Desse modo, a interdisciplinaridade entre matemática e computação constitui uma ponte essencial entre teoria e prática, contribuindo diretamente para o desenvolvimento do raciocínio lógico, da capacidade analítica dos estudantes e da formação de profissionais mais bem preparados para os desafios impostos pelo avanço científico e tecnológico.

REFERÊNCIAS

ATKINSON KE. **An introduction to numerical analysis**. 2nd ed. New York: John Wiley & Sons, 2008.

BRASIL. Conselho Nacional de Saúde. Resolução nº 466/2012 sobre pesquisa envolvendo seres humanos. Brasília, 2012.

BURDEN RL, FAIRES JD. **Numerical analysis**. 9th ed. Boston: Brooks Cole, 2011; 600p.

CHAPRA SC, CANALE RP. **Métodos numéricos para engenharia**. 7th ed. São Paulo: McGraw-Hill, 2015.

GIL AC. **Métodos e técnicas de pesquisa social**. 6th ed. São Paulo: Atlas, 2017.

ISO/IEC. ISO/IEC 14882:2017 – **Programming languages — C++. Geneva**: ISO, 2017.

KINCAID RC, CHENEY WR. **Numerical analysis: mathematics of scientific computing**. 3rd ed. Pacific Grove: Brooks/Cole, 2002

LAKATOS EM, MARCONI MA. **Fundamentos de metodologia científica**. 7th ed. São Paulo: Atlas, 2017;

MATHWORKS. MATLAB documentation. Natick, MA: The MathWorks Inc., 2024; 500p.

OLIPHANT TE. **Python for scientific computing. Computing in Science & Engineering**, 2007; 9(3): 10-20.

SILVA JF, et al. **Performance evaluation of numerical methods in scientific computing. Journal of Computational Science**, 2019; 34: 150-160.

VAN ROSSUM G, DRAKE FL Jr. **Python language reference manual**. 3rd ed. Scotts Valley: CreateSpace, 2009.