

Aplicando os Módulos Numpy e Matplotlib em Python para o Ensino de Cálculo Numérico em Cursos de Física e Engenharia

DOI: 10.37702/2175-957X.COBENGE.2023.4575

Sérgio Ricardo Xavier da Silva - srxsilva@uneb.br
Universidade do Estado da Bahia

Resumo: *As mudanças no setor tecnológico provocam modificações constantes nas metodologias de ensino e nos projetos pedagógicos, a modernização das disciplinas é necessária para acompanhar o desenvolvimento tecnológico atual. As consequências dessas mudanças no setor acadêmico são maiores nas disciplinas diretamente relacionadas com a Computação, como é o caso de Cálculo Numérico, disciplina comum nos cursos de Física e Engenharia. Este artigo apresenta uma avaliação da utilização de algoritmos desenvolvidos em linguagem Python para o aprendizado de métodos numéricos com o objetivo de tornar o aprendizado mais estimulante e eficiente dessa disciplina.*

Palavras-chave: *Ensino na Engenharia, Ensino na Física, Cálculo Numérico, Python.*

APLICANDO OS MÓDULOS NUMPY E MATPLOTLIB EM PYTHON PARA O ENSINO DE CÁLCULO NUMÉRICO EM CURSOS DE FÍSICA E ENGENHARIA

1 INTRODUÇÃO

Chapra e Canale (2016) explanam que métodos numéricos representam técnicas matemáticas empregadas na obtenção de soluções numéricas para uma série de problemas, tais como aqueles encontrados na Física e Engenharia, especialmente quando as soluções analíticas são inviáveis. A solução analítica pode ser entendida como uma resposta matematicamente precisa que é formulada na forma de uma expressão matemática, correlacionada às variáveis do problema investigado. Por outro lado, a solução numérica proporciona uma aproximação numérica para a solução do problema em discussão, como discutido por Burden, Faires e Burden (2016).

Conforme Filho (2016) apresenta, os métodos numéricos são atualmente implementados em computadores, equipamentos capazes de executar uma enorme quantidade de operações lógicas e aritméticas de maneira repetitiva em um intervalo de tempo reduzido.

Operações extensivas envolvendo matrizes e vetores, raízes de equações algébricas e transcendentais, interpolação, integração, equações diferenciais ordinárias e ajuste de curvas figuram como tópicos fundamentais de disciplinas de Cálculo Numérico. Desta forma, é indiscutível que estudantes de Física e Engenharia devem dominar uma variedade de métodos numéricos para a resolução de diversos tipos de problemas.

No entanto, tem-se constatado que a aprendizagem desses conteúdos pode se mostrar insatisfatória para muitos estudantes, especialmente quando tecnologias computacionais não são incorporadas como ferramentas de apoio ao ensino em sala de aula. Esta lacuna no aprendizado é evidenciada em alunos ao longo de todos os níveis de educação, desde a educação básica até o ensino superior, conforme apontam Notare e Behar (2009).

Serres e Basso (2009) destacam a necessidade de auxiliar os estudantes na construção de estratégias efetivas de aprendizagem. No contexto específico do Cálculo Numérico, os computadores deveriam desempenhar um papel crucial para apoiar os estudantes na aquisição desse conhecimento. No entanto, Quadros e Martins (2005) ressaltam que o simples uso de novas tecnologias pode não ser suficiente, já que essas ferramentas podem ser empregadas sem que suas características específicas sejam exploradas devidamente.

Assim, surgem importantes questões: Como ministrar o Cálculo Numérico, presencialmente ou remotamente, de forma eficiente com o auxílio de computadores? Como potencializar o interesse dos alunos pelo conteúdo abordado? Essas indagações conduziram à proposição de desenvolver códigos em Python utilizando os módulos NumPy e Matplotlib direcionados ao ensino de Cálculo Numérico.

Conforme postulado por Chinaglia et al (2022), o domínio das habilidades de programação permite que os estudantes automatizem uma série de tarefas, notadamente aquelas que são complexas e repetitivas. Essa competência, permite que os estudantes utilizem a Computação para outros propósitos, otimizando e agilizando tarefas cotidianas.

Nesse sentido, o propósito deste artigo é avaliar o impacto da utilização dos módulos NumPy e Matplotlib, desenvolvidos na linguagem Python, como suporte à aprendizagem dos conteúdos de Cálculo Numérico, em cursos superiores presenciais, híbridos ou integralmente à distância, de Física e Engenharia, fazendo uma mudança gradual da linguagem Scilab, utilizada atualmente, para a linguagem Python.

O presente trabalho está organizado da seguinte maneira. A seção 2 traça uma breve descrição sobre os principais módulos para o ensino de Cálculo Numérico: Numpy e Matplotlib. A seção 3 apresenta a pesquisa realizada com o uso do Python nas aulas de Cálculo Numérico. A seção 4 apresenta os resultados da pesquisa. Por fim, a seção 5 conclui o artigo apresentando as considerações e perspectivas futuras.

2 MÓDULOS PYTHON PARA CÁLCULO NUMÉRICO

O Python é uma linguagem de programação de alto nível que oferece uma extensa quantidade de bibliotecas para processamento de dados. O módulo Matplotlib é uma dessas bibliotecas, permitindo a criação de gráficos bi e tridimensionais de alta qualidade. O Numpy é outra biblioteca importante para Cálculo Numérico, permitindo a manipulação de vetores e matrizes.

Um exemplo prático de como essas ferramentas computacionais podem ser utilizadas para melhorar o ensino de Cálculo Numérico é a visualização de funções. Com o módulo Matplot.lib, é possível criar gráficos de funções em uma variedade de estilos e formatos. Isso, evidentemente, é útil para ilustrar conceitos importantes em Cálculo Numérico, como análise de funções, limites, derivadas, integrais e interpolação.

Além disso, como o Numpy pode ser utilizado para realizar operações matemáticas em matrizes e vetores, conceitos importantes em Cálculo Numérico, como a resolução de sistemas lineares e a decomposição de matrizes, podem ser trabalhados em sala de aula de forma mais eficiente.

Existem várias IDEs¹ disponíveis para programar em Python, PyCharm, Jupyter Notebook, Spyder, Visual Studio Code e o Google Colaboratory, são alguns exemplos.

O Google Colaboratory é uma plataforma gratuita baseada na nuvem que permite aos usuários escrever e executar códigos Python em um ambiente de notebook Jupyter. Conhecido como Google Colab, esta IDE é executada no navegador e é hospedada nos servidores do Google, permitindo aos usuários trabalharem em projetos de ciência de dados, aprendizado de máquina e inteligência artificial sem precisar configurar um ambiente de desenvolvimento em seus próprios computadores.

O Google Colab vem com vários recursos, incluindo bibliotecas pré-instaladas, acesso a GPUs e TPUs gratuitos, integração com o Google Drive para salvar e compartilhar notebooks e colaboração com outros usuários. Ele é uma poderosa ferramenta para estudantes, pesquisadores e desenvolvedores que desejam explorar e criar algoritmos em um ambiente de nuvem gratuito. Devido a todas as vantagens citadas anteriormente, o Google Colaboratory foi a IDE escolhida para o desenvolvimento desta pesquisa.

¹ IDE é a sigla para Integrated Development Environment, que em português pode ser traduzido como Ambiente de Desenvolvimento Integrado. É um software que oferece uma série de ferramentas para desenvolvedores de software, visando facilitar e agilizar o processo de desenvolvimento de programas.

2.1 Módulo Numpy

A biblioteca NumPy, da linguagem de programação Python, é uma ferramenta imprescindível no ensino do Cálculo Numérico. Projetada especificamente para operações matemáticas e científicas, a biblioteca NumPy fornece uma interface fácil de usar e eficiente para lidar com vetores e matrizes de dados. Nesse contexto, discutiremos a aplicação do NumPy no ensino de Cálculo Numérico, ilustrando com exemplos de código.

Consideremos primeiro, a solução de sistemas de equações lineares, uma tarefa fundamental em muitos problemas de Cálculo Numérico. Com o NumPy, resolver um sistema de equações lineares pode ser simplificado usando a função `numpy.linalg.solve`. Por exemplo, para resolver o sistema de equações $3x + 2y = 2$ e $x - 2y = 1$, pode-se escrever o código apresentado na Figura 1.

Figura 1 – Código para solução de um sistema de equações.

```
import numpy as np

# Coeficientes à esquerda das igualdades
A = np.array([[3, 2], [1, -2]])

# Valores à direita das igualdades
b = np.array([2, 1])

# Solução do sistema
sol = np.linalg.solve(A, b)
sol
```

Fonte: Próprio do autor.

Outro aspecto fundamental do Cálculo Numérico é a manipulação de matrizes e vetores. O NumPy permite operações eficientes em matrizes e vetores com sintaxe simples. Por exemplo, a multiplicação de matrizes, um conceito básico em álgebra linear, pode ser realizada com a função `numpy.dot`, conforme pode-se observar na Figura 2. Desta forma, o aluno concentra-se no método numérico para solução do problema.

Figura 2 – Código para multiplicação de matrizes utilizando a função `numpy.dot`.

```
import numpy as np

# Duas matrizes 2x2
A = np.array([[1, 2], [3, 4]])
B = np.array([[5, 6], [7, 8]])

# Multiplicação de matrizes
C = np.dot(A, B)
C
```

Fonte: Próprio do autor.

O NumPy também é útil para a implementação de métodos numéricos, como a integração numérica, dentre outros. Por exemplo, o método trapezoidal, um algoritmo básico de integração numérica, pode ser implementado usando o NumPy conforme pode-se observar na Figura 3.

Figura 3 – Integração numérica pelo método trapezoidal.

```
import numpy as np

# Função a ser integrada
def f(x):
    return np.sin(x)

# Intervalo de integração e número de trapezóides
a, b, N = 0, np.pi, 1000

# Criação dos x valores
x = np.linspace(a, b, N+1)

# Cálculo da integral
integral = np.trapz(f(x), x)
integral
```

Fonte: Próprio do autor.

Esses exemplos ilustram como a biblioteca NumPy pode ser aplicada de maneira eficiente e intuitiva no ensino do Cálculo Numérico. Através de seu uso, os estudantes podem ganhar uma compreensão mais concreta dos conceitos matemáticos enquanto desenvolvem habilidades de programação de computadores.

2.2 Módulo Matplotlib

O módulo Matplotlib da linguagem de programação Python é uma ferramenta importante no ensino do Cálculo Numérico. Ele fornece uma ampla gama de funções para a criação de gráficos bi e tridimensionais, estáticos, animados e interativos de alta qualidade, tornando-o essencial para a visualização e interpretação de resultados matemáticos.

Um exemplo prático de como o módulo Matplotlib pode ser usado para melhorar o ensino de Cálculo Numérico é a visualização de funções. Com o Matplotlib, é possível criar gráficos de funções em uma enorme variedade de estilos e formatos. Isso pode ser útil para ilustrar conceitos importantes em Cálculo Numérico, como análise de funções, limites, derivadas, integrais e interpolação.

Por exemplo, considere a função $\text{sen}(x)$, utilizando o Matplotlib, podemos criar um gráfico desta função no intervalo $[0, 2\pi]$, como pode-se observar o código na Figura 4, e o gráfico gerado na Figura 5.

Figura 4 – Utilização do módulo Matplotlib para plotagem de gráficos de funções.

```
import numpy as np
import matplotlib.pyplot as plt

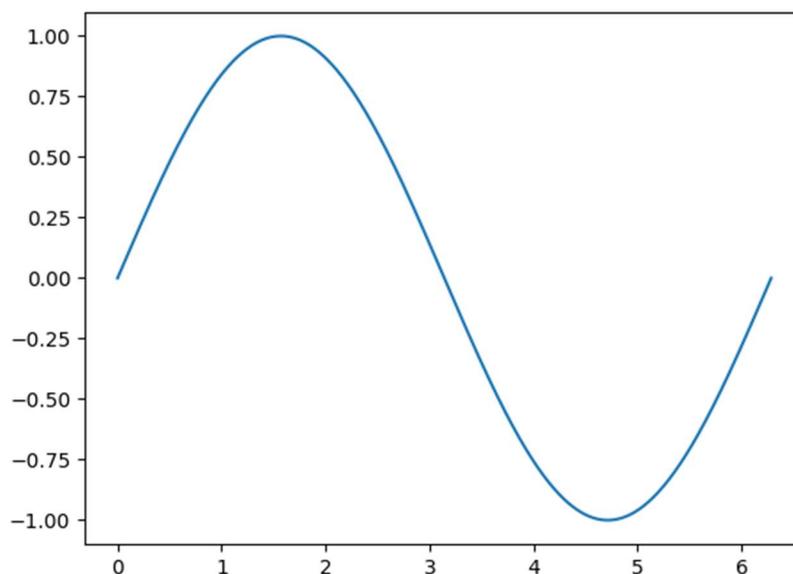
# Criação do array de x valores
x = np.linspace(0, 2*np.pi, 1000)

# Cálculo dos y valores correspondentes
y = np.sin(x)

# Plot da função
plt.plot(x, y)
plt.show()
```

Fonte: Próprio do autor.

Figura 5 – Gráfico gerado a partir do Matplotlib.



Fonte: Próprio do autor.

Além disso, o Matplotlib também pode ser usado para visualizar a solução de equações diferenciais ordinárias, uma importante área do Cálculo Numérico. Por exemplo, para visualizar a solução da equação diferencial $\frac{dy}{dx} = y$, pode-se utilizar o código apresentado na Figura 6.

Figura 6 – Código em Python para solução de equações diferenciais ordinárias.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint

# Definição da função que representa a equação diferencial
def f(y, x):
    return y

# Condição inicial e array de x valores
y0 = 1.0
x = np.linspace(0, 2, 100)

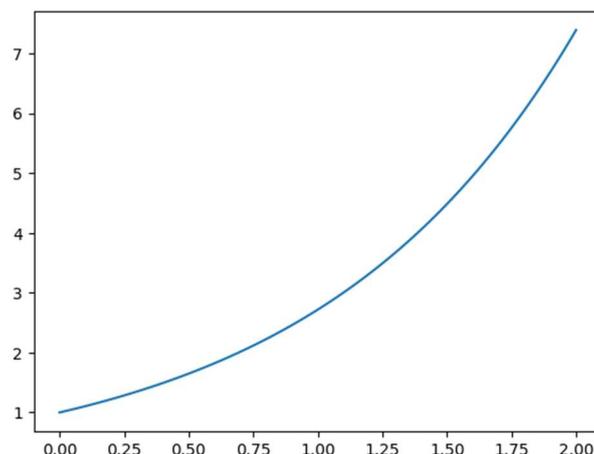
# Solução da equação diferencial
y = odeint(f, y0, x)

# Plot da solução
plt.plot(x, y)
plt.show()
```

Fonte: Próprio do autor.

O gráfico da solução da equação diferencial ordinária pode ser observado na Figura 7, a seguir.

Figura 7 – Gráfico da solução da equação diferencial ordinária gerado pelo módulo Matplotlib.



Fonte: Próprio do autor.

Esta biblioteca também possui uma funcionalidade poderosa que permite aos usuários incorporar expressões em LaTeX² para aprimorar a qualidade dos gráficos

² LaTeX é um sistema de preparação de documentos de alta qualidade, amplamente utilizado para a composição de textos matemáticos e científicos. Ele é baseado no sistema de tipografia TeX e fornece uma linguagem de marcação que permite aos autores estruturar e formatar seus documentos com precisão.

produzidos. Isso é extremamente útil para incluir equações matemáticas complexas ou simplesmente para formatar títulos, eixos e legendas de maneira mais elegante e precisa.

Para isso, basta incluir o texto em formato LaTeX dentro de strings, envolvendo-o com marcadores de cifrão (\$), desta forma, com a ajuda do interpretador matemático LaTeX e a biblioteca de renderização de texto Mathtext, o Matplotlib processa essa string e exibe a equação formatada corretamente.

Estes exemplos evidenciam a capacidade do Matplotlib de auxiliar na compreensão dos conceitos de Cálculo Numérico através da visualização gráfica. Com o seu uso, os estudantes podem ter uma maior intuição sobre as operações e conceitos que estão estudando, além de adquirir habilidades em programação e visualização de dados.

3 O PYTHON NAS AULAS DE CÁLCULO NUMÉRICO

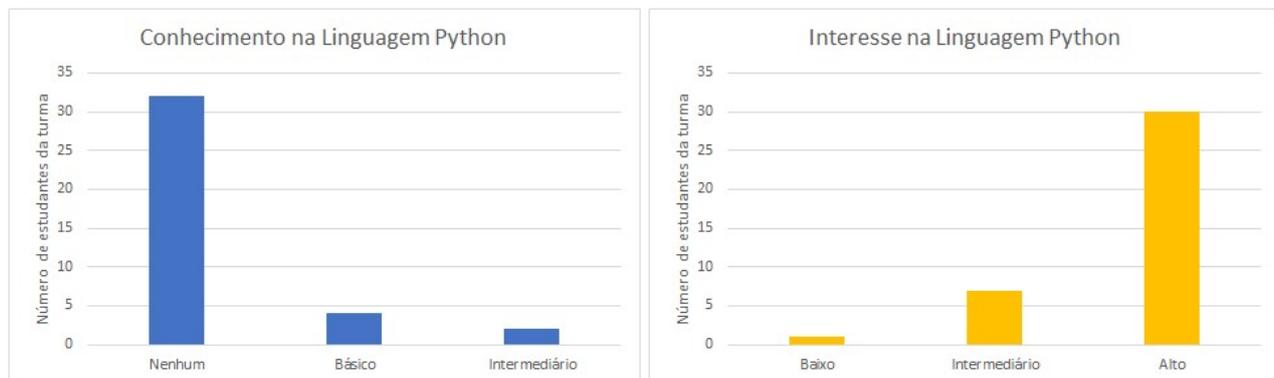
Esta pesquisa teve como finalidade o desenvolvimento de algoritmos computacionais em Python, aplicado às aulas de Cálculo Numérico em duas turmas, A e B, de Cálculo Numérico, cada uma com 38 alunos matriculados e com frequência regular. Todas as turmas utilizam o Scilab como ferramenta didática. A pesquisa ocorreu em três etapas, a primeira etapa consistiu na apresentação do Python e seus módulos Numpy e Matplotlib aos alunos, esclarecendo as dúvidas pertinentes à sua utilização, evitando possíveis dificuldades na utilização desta linguagem.

Esta primeira etapa, ocorreu após uma sondagem realizada no início do semestre, com os 38 alunos matriculados na turma A de Cálculo Numérico. Todos os estudantes desta turma participaram da pesquisa, e responderam um questionário com as seguintes perguntas:

- Atualmente existem inúmeras linguagens de programação tais como Pascal, C, C++, C#, Java, e Python, por exemplo. Você tem algum conhecimento em Python?
- Qual o seu interesse no uso da linguagem Python para o desenvolvimento de programas aplicados na disciplina de Cálculo Numérico?

Os resultados desta sondagem, sobre as linguagens de programação, são mostrados na Figura 8.

Figura 8 – Questionário aplicado no início do semestre aos alunos da disciplina de Cálculo Numérico.



Fonte: Próprio do autor.

A segunda etapa do trabalho inicialmente consistiu na definição do conteúdo a ser abordado durante o desenvolvimento dos códigos em Python para a turma A. O conteúdo escolhido foi Integração Numérica. É importante salientar que a disciplina é desenvolvida utilizando Scilab e a seleção de um conteúdo, dentro de todo o programa da disciplina, teve como objetivo comparar o uso dessas linguagens distintas como ferramentas didáticas no ensino da disciplina. Isso foi feito comparando os resultados dos questionários aplicados nas duas turmas, bem como os resultados das avaliações.

A terceira etapa foi realizada após a definição do conteúdo, sendo modelados os algoritmos dos métodos: regra dos trapézios, 1ª e 2ª regras de Simpson, extrapolação de Richardson, integração dupla e quadratura gaussiana. A modelagem visou apresentar os métodos de forma clara, objetiva e organizada, o que facilitou a posterior implementação utilizando a linguagem Python. Arquivos (.ipynb)³ foram criados, um para cada método, contendo as implementações dos métodos escolhidos. A implementação foi executada conforme os algoritmos previamente modelados, com instruções de saída em todas as etapas intermediárias.

A título de ilustração, na Subseção 2.1, foi apresentado um código em Python para integração numérica utilizando o método trapezoidal, após apresentação dos demais métodos, os alunos implementaram o método da extrapolação de Richardson para solução do mesmo problema. O código desenvolvido pode ser observado na Figura 9 e o gráfico da solução, na Figura 10.

Figura 9 – Código em Python para integração numérica utilizando o método da extrapolação de Richardson.

```
import numpy as np
import matplotlib.pyplot as plt

def f(x):
    return np.sin(x) # Função trigonométrica de exemplo

def Extrapolacao_de_Richardson(a, b, n):
    h = (b - a) / (2**n)
    x = np.linspace(a, b, 2**n + 1)
    y = f(x)

    integral = 0
    for i in range(0, 2**n, 2):
        integral += h/3 * (y[i] + 4*y[i+1] + y[i+2])

    return integral

# Configurações para o comando plot
a = 0
b = np.pi
valores_de_n = np.arange(1, 10)
valores_da_integral = []

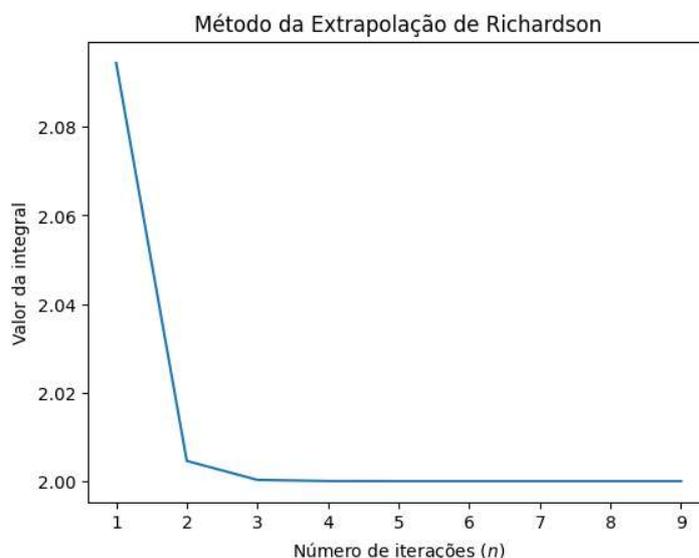
# Cálculo a integral para diferentes valores de n
for n in valores_de_n:
    integral = Extrapolacao_de_Richardson(a, b, n)
    valores_da_integral.append(integral)

# Plotagem do valor da integral em função de n
plt.plot(valores_de_n, valores_da_integral)
plt.xlabel('Número de iterações ($n$)')
plt.ylabel('Valor da integral')
plt.title('Método da Extrapolação de Richardson')
plt.show()
```

Fonte: Próprio do autor.

³ Um arquivo IPYNB é um documento de caderno usado pelo Jupyter Notebook, um ambiente computacional interativo projetado para ajudar os cientistas a trabalhar com a linguagem Python e seus dados.

Figura 10 – Plotagem do valor da integral utilizando o método da extrapolação de Richardson.



Fonte: Próprio do autor.

Depois de desenvolver e testar os códigos, realizaram-se aulas práticas subsequentes à apresentação dos conteúdos. Para cada conteúdo discutido em sala, foram criadas listas de exercícios que deveriam ser resolvidos inicialmente à mão e posteriormente no computador, utilizando os códigos desenvolvidos em Python.

Ao término do semestre, os estudantes da turma A preencheram uma pesquisa online, visando mensurar o grau de motivação para o uso de Python em sala de aula. A ferramenta utilizada para a pesquisa online foi o formulário eletrônico do Google Drive.

Todos os 38 alunos matriculados na disciplina participaram da pesquisa. O formulário foi preenchido de forma anônima e subjetiva. Apenas uma pergunta deveria ser respondida: Expresse sua avaliação sobre o uso dos algoritmos em Python no processo de ensino e aprendizagem de Cálculo Numérico.

4 RESULTADOS

Os algoritmos desenvolvidos em Python proporcionam uma exposição detalhada das operações indispensáveis para a compreensão de cada método em estudo. Eles simulam uma solução manual, oferecendo cálculos progressivos que auxiliam o aluno no processo de solução de problemas ao permitir a verificação de cada etapa da resolução.

Em resumo, o processo de aprendizado requer dos estudantes, em primeiro lugar, a identificação dos dados de entrada para o problema em questão. Em seguida, eles devem observar as saídas intermediárias antes de analisar a saída final.

A análise das respostas aos questionários aplicados a todos os estudantes indicou um consenso unânime sobre a relevância dessa abordagem. Ou seja, a importância de integrar teoria e prática usando Python foi reconhecida amplamente.

Também foram aplicadas avaliações formais, onde os alunos deveriam desenvolver códigos para o cálculo da integral de funções a partir dos métodos numéricos estudados. Estas avaliações foram aplicadas nas turmas A e B, sendo que apenas a turma A utilizou

os módulos Numpy e Matplotlib em Python, a turma B utilizou o Scilab para desenvolver os algoritmos. A média das notas da turma A (8,07), foi levemente maior do que a da turma B (7,76).

A importância inicial da implementação dos códigos criados no âmbito da aula de Cálculo Numérico se refletiu no resultado acadêmico final dos alunos. Na turma A, de um conjunto de trinta e oito estudantes, apenas seis tiveram que fazer a avaliação final, o que significa que 32 deles conseguiram aprovação através da média. Em contrapartida, na turma B, 29 estudantes conseguiram evitar a necessidade da avaliação final.

5 CONSIDERAÇÕES FINAIS

Nesta pesquisa, discutiu-se o desenvolvimento de algoritmos em Python para auxiliar no ensino de Cálculo Numérico. O estudo atual ofereceu justificativa e evidência empírica para a importância da incorporação de tecnologias computacionais no ensino de Cálculo Numérico. A prática de desenvolver algoritmos em Python pelos estudantes resultou em aumento do interesse, melhora no desempenho final e um ambiente de aprendizado mais estimulante e envolvente. Resumidamente, houve uma melhoria na qualidade do ensino e aprendizado em Cálculo Numérico.

A experiência adquirida ressalta que o papel do professor de Cálculo Numérico não deve ser restrito à mera exposição de conteúdos em sala de aula. Pesquisas estão em andamento na disciplina de Cálculo Numérico com o objetivo de detalhar uma série de atividades experimentais em uma futura publicação. Tais atividades visam comprovar a aplicabilidade dos códigos implementados em Python em mais tópicos da disciplina, propondo desta forma, uma migração gradual da linguagem Scilab para o Python.

REFERÊNCIAS

BURDEN, R. L.; FAIRES, D. J.; BURDEN, A. M. **Análise Numérica - Tradução da 10ª Edição Norte-Americana**. Cengage Learning, 2016.

CANALE, R. P.; CHAPRA, S. C. **Métodos Numéricos Para Engenharia**. 7ª. ed. Amgh Editora, 2016.

CHINAGLIA, Eliane F. *et al.* Python como ferramenta didática em disciplinas de Física para graduação em Engenharia. In: L Congresso Brasileiro de Educação em Engenharia, 2022, Evento Online. **Anais**. Disponível em: http://www.abenge.org.br/sis_submetidos.php?acao=abrir&evento=COBENGE22&codigo=COBENGE22_00300_00003862.pdf. Acesso em 04 jun.2023.

FILHO, A. A. D. **Fundamentos de Cálculo Numérico**. Bookman, 2016.

NOTARE, P.; BEHAR, M. R. Aprendizagem e comunicação matemática em ambientes virtuais: Uma experiência com o cálculo diferencial. **XX Simpósio Brasileiro de Informática na Educação**, Florianópolis, 2009.

QUADROS, T.; MARTINS, J. A prática interdisciplinar em programas de educação a distância num cenário de novas tecnologias da informação e comunicação. **XVI Simpósio Brasileiro de Informática na Educação**, 2005.

SERRES, F. F.; BASSO, M. V. A. Diários virtuais – Uma ferramenta de comunicação social para a autoria e aprendizagem de Matemática. **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)**, 2009.

APPLYING THE NUMPY AND MATPLOTLIB MODULES IN PYTHON FOR THE TEACHING OF NUMERICAL CALCULUS IN PHYSICS AND ENGINEERING COURSES

Abstract: *Changes in the technology sector provoke constant modifications in teaching methodologies and pedagogical projects, modernization of subjects is necessary to keep up with current technological development. The consequences of these changes in the academic sector are greatest in subjects directly related to Computing, such as Numerical Calculation, a common subject in Physics and Engineering courses. This paper presents an evaluation of the use of algorithms developed in Python language for the learning of numerical methods with the aim of making the learning more stimulating and efficient for this subject.*

Keywords: *Teaching in Engineering, Teaching in Physics, Numerical Calculus, Python.*