

Artigo

LABPYTHON: Os modelos de Malthus e Verhulst aplicados ao estudo da dinâmica populacional da cidade de Rio Grande

Biller, J. R.^{1,*}, Rodriguez, B. D. A.² e Poffal, C. A.³

¹ Instituto de Matemática, Estatística e Física - Universidade Federal do Rio Grande; jhonatan.biller@furg.br

² Instituto de Matemática, Estatística e Física - Universidade Federal do Rio Grande; barbararodriguez@furg.br

³ Instituto de Matemática, Estatística e Física - Universidade Federal do Rio Grande; poffal@gmail.com

* Correspondência: jhonatan.biller@furg.br

Received: 18/01/2024; Accepted: 25/01/2024; Published: 31/01/2024

Resumo: Este trabalho apresenta soluções para o problema de crescimento populacional, aplicado ao município de Rio Grande - RS, entre os anos de 2010 e 2012, formulado pelos modelos de dinâmica populacional de Malthus e Verhulst. Utiliza-se a plataforma Jupyter Notebook, junto a linguagem de programação Python e suas bibliotecas SymPy, NumPy e Matplotlib, para implementar os métodos de solução numérica de Euler, Euler Modificado e Runge-Kutta de 4ª ordem, exibir e documentar os resultados obtidos. Comparam-se os dados reais para os anos presents na formação do problema com os resultados obtidos pelos modelos populacionais. Faz-se, ainda, uma estimativa para o ano de 2022, que é comparada aos dados fornecidos pelo Instituto Brasileiro de Geografia e Estatística (IBGE) para esse mesmo ano. Os resultados apresentados são satisfatórios, com erros inferiores a 4%.

Palavras-Chave: Python; Métodos Numéricos; Dinâmica Populacional.

Article

LABPYTHON: Malthus and Verhulst models applied to the study of population dynamics in the city of Rio Grande

Biller, J. R.^{1,*}, Rodriguez, B. D. A.² e Poffal, C. A.³

¹ Institute of Mathematics, Statistics, and Physics - Federal University of Rio Grande; barbararodriguez@furg.br

² Institute of Mathematics, Statistics, and Physics - Federal University of Rio Grande; poffal@gmail.com

³ Institute of Mathematics, Statistics, and Physics - Federal University of Rio Grande; jhonatan.biller@furg.br

* Correspondence: jhonatan.biller@furg.br

Abstract: This work presents solutions to the population growth problem applied to the municipality of Rio Grande - RS, between the years 2010 and 2012, formulated by the population dynamics models of Malthus and Verhulst. The Jupyter Notebook platform is employed, along with the Python programming language and its libraries SymPy, NumPy, and Matplotlib, to implement the numerical solution methods of Euler, Modified Euler, and 4th-order Runge-Kutta, display, and document the obtained results. Real data for the years present in the problem formulation are compared with the results obtained by the population models. Additionally, an estimate is made for the year 2022, which is compared to the data provided by the Brazilian Institute of Geography and Statistics (IBGE) for the same year. The presented results are satisfactory, with errors below 4%.

Keywords: Python; Numerical Methods; Population Dynamics.

1. Introdução

O estudo da dinâmica populacional é fundamental para o planejamento econômico de uma região, pois possibilita projeções demográficas futuras. A partir do século XVIII, o estatístico e economista Thomas Malthus, interessado na relação entre o crescimento da população humana e a produção de comida, desenvolveu um modelo matemático para estudar o crescimento populacional mundial (Bacaër, 2011). Malthus publicou que o poder de crescimento da população humana é maior que o poder da terra para produzir substâncias para o homem. Propôs que essa população tende a crescer em progressão geométrica, dando origem ao Modelo de Malthus (Mill, 1996). Em 1838, observando que o crescimento populacional, segundo o modelo proposto por Malthus, cresce indefinidamente ao longo do tempo, Pierre-François Verhulst apontou que o aumento da população está limitado pelo tamanho e fertilidade de um meio biológico (Bacaër, 2011), e, em 1847, apresentou uma equação diferencial que descrevia a população considerando as características do meio (Modelo de Verhulst).

As soluções matemáticas para as Equações Diferenciais, que modelam a dinâmica populacional, podem ser resolvidas de duas maneiras: analítica ou numericamente. Na primeira, apresenta-se uma solução funcional para o problema (Boyce, 2006). Já a segunda corresponde a discretizar o domínio da função associada à equação diferencial (Boyce, 2006), calculando-se aproximações para a solução exata. As técnicas de solução numérica são amplamente implementadas computacionalmente, uma vez que soluções exatas, em muitos casos, demandam elevado custo em relação ao tempo de execução, ou ainda, impraticáveis. Dentre as linguagens de programação, com potencial para lidar com as operações matemáticas necessárias, encontra-se o Python. O Python é uma linguagem de programação de código aberto de alto nível. Popularizou-se pela sua simplicidade e eficiência para atuar em diferentes áreas (Python, 2023), tais como: Desenvolvimento Web; Interfaces Gráficas; Programação de Softwares e Computação Científica. Possui, ainda, bibliotecas terceirizadas capazes de resolver ou auxiliar a implementação de técnicas de solução.

Dentro deste contexto, este trabalho tem como objetivo estudar a dinâmica populacional do município de Rio Grande - RS, a partir dos modelos de Malthus e Verhulst. As soluções são obtidas a partir do emprego dos métodos numéricos de Euler, Euler Modificado e Runge-Kutta de 4ª ordem. Foram utilizados dados populacionais disponibilizados no DATASUS (Datusus, 2022), que representa a população real entre os anos de 2010 e 2012, a fim de avaliar o crescimento populacional do município e contribuir para uma melhor compreensão da dinâmica do desenvolvimento populacional urbano da região. Os algoritmos são implementados por meio da linguagem de programação Python. Empregam-se, ainda, funções internas, disponibilizadas pelas bibliotecas Python junto ao Jupyter Notebook, para analisar, documentar, exibir e estabelecer estudos comparativos entre os resultados obtidos. Para atingir as metas delineadas, esta pesquisa segue a estrutura a seguir: Na seção 2, explora-se a base matemática, abordando tanto a formulação dos modelos populacionais quanto as técnicas numéricas empregadas na resolução. A seção 3 destaca os resultados das simulações populacionais propostas. Por sua vez, a seção 4 oferece uma avaliação sobre a linguagem de programação Python e a interpretação dos resultados alcançados. Por fim, a seção 5 apresenta as conclusões deste trabalho.

2. Fundamentação Matemática

Na presente seção, é apresentada a fundamentação matemática essencial para a compreensão aprofundada das dinâmicas populacionais. Exploram-se dois modelos fundamentais: o Modelo Malthusiano e o Modelo de Verhulst (Bacaër, 2011). Após a exposição desses modelos teóricos, abordam-se os métodos numéricos de passo único (Chapra, 2011), fundamentais para a resolução eficaz de equações diferenciais ordinárias, incluindo o Método de Euler, o Euler Modificado e o Método Runge-Kutta de 4ª ordem.

2.1. Modelos de Dinâmica Populacional

A compreensão da dinâmica populacional é fundamental para analisar as variações no tamanho e na estrutura de uma população ao longo do tempo. Modelos matemáticos desempenham um papel crucial nessa análise, oferecendo ferramentas para prever e compreender os padrões de crescimento populacional. A seguir, são apresentadas as equações diferenciais que descrevem os modelos de Malthus e Verhulst (Bacaër, 2011).

2.1.1. Modelo de Malthus

Malthus (1798) emergiu como um dos precursores na elaboração de modelos matemáticos ao desenvolver o que ficou conhecido como o modelo Malthusiano. Este modelo aborda a interação entre o crescimento populacional e a capacidade de sustentação dos recursos disponíveis. A teoria de Malthus sustentava que a população humana tinha o potencial de crescer de maneira exponencial, enquanto a produção de alimentos aumentava em uma progressão

aritmética, mais lentamente. Ele antecipou que, se o crescimento populacional não fosse controlado, surgiriam escassez de alimentos e crises como fome e miséria resultantes dessa discrepância entre crescimento populacional e produção de alimentos. O modelo de Malthus é formulado em termos de equações diferenciais por:

$$\frac{dP}{dt} = \kappa P, \quad (1)$$

onde κ é a taxa de crescimento populacional e $P = P(t)$ é a população no instante t .

2.1.2. Modelo de Verhulst

Com base nas análises de Malthus e Quetelet (Bacaër, 2010) sobre a dinâmica populacional, Pierre François Verhulst apresentou, em 1838, seu estudo intitulado "Notas sobre a Lei de Crescimento Populacional". Nesta pesquisa, Verhulst propôs que o crescimento populacional, conforme concebido pelo modelo malthusiano, deveria encontrar um limite determinado pelas dimensões e fertilidade específicas do país em foco. Esse modelo é formulado por:

$$\frac{dP}{dt} = \kappa P \left(1 - \frac{P}{K}\right), \quad (2)$$

onde K é a capacidade suporte do meio biológico.

2.2. Métodos Numéricos de Passo Único

Na análise de fenômenos dinâmicos, a resolução de equações diferenciais desempenha um papel vital na previsão do comportamento futuro de sistemas complexos. Nesse contexto, métodos numéricos de passo único (Chapra, 2011) surgem como ferramentas essenciais, oferecendo abordagens eficazes para discretizar e resolver equações problemas do valor inicial da forma

$$\begin{cases} \frac{dP}{dt} = f(t, P), \\ P(t_0) = P_0. \end{cases} \quad (3)$$

Os métodos conhecidos como "de passo único", calculam um novo valor P_{i+1} . com base no estado atual do sistema P_i , na inclinação φ , e em um passo h , isto é,

$$P_{i+1} = P_i + \varphi h, \quad (4)$$

onde o passo h é uma constante utilizada para subdividir o domínio da variável independente em segmentos menores, baseando-se no valor atual da variável independente t_i para calcular o próximo valor t_{i+1} , dado por

$$t_{i+1} = t_i + h. \quad (5)$$

Cada segmento corresponde a um intervalo de tempo, e a equação diferencial é solucionada em cada ponto desse intervalo por meio de uma aproximação discreta. A seguir, definem-se três métodos fundamentais: o Método de Euler, o Euler Modificado e o Método de Runge-Kutta de 4ª ordem.

2.2.1. Método de Euler

O Método de Euler, desenvolvido por Leonhard Euler no século XVIII, é um exemplo notável de um método de passo único. Sua formulação simples envolve o uso da própria equação diferencial (3) como inclinação para estimar o ponto P_{i+1} com base no estado atual do sistema t_i e P_i , com finalidade de construir uma solução numérica para equações diferenciais ordinárias de primeira ordem. A fórmula de atualização do Método de Euler é dada por

$$P_{i+1} = P_i + f(t_i, P_i)h. \quad (6)$$

2.2.2. Método de Euler Modificado

Assim como o Método de Euler, o Método de Euler Modificado opera com uma abordagem de passo único, discretizando a solução em incrementos sucessivos ao longo do tempo. No entanto, aprimora a precisão ao utilizar uma média ponderada das inclinações no início e no final do intervalo de passo. As fórmulas de atualização para esse método são dadas por

$$\begin{aligned} \text{Preditor: } P_{i+1}^0 &= P_i + f(t_i, P_i)h. \\ \text{Corretor: } P_{i+1} &= P_i + (h/2) \left(f(t_i, P_i) + f(t_{i+1}, P_{i+1}^0) \right). \end{aligned} \quad (7)$$

Nesse método de aproximação para o problema de valor inicial, a equação corretiva está atrelada à equação preditora em cada iteração, seguindo uma abordagem de dois estágios. Durante cada passo, são obtidas duas estimativas da inclinação da função, uma no início e outra no final do intervalo. A média dessas inclinações é então utilizada para aprimorar a estimativa da solução ao longo do intervalo.

2.2.3. Método Runge-Kutta de 4ª ordem

O Método de Runge-Kutta de 4ª ordem (RK4) é um método numérico utilizado para resolver equações diferenciais ordinárias. Ele se destaca por sua precisão de quarta ordem e é especialmente eficaz para resolver problemas de valor inicial. O RK4 opera em um processo de quatro passos, proporcionando uma aproximação mais precisa da solução em comparação com métodos de ordem inferior. Nesse método, a quantidade P_{i+1} é calculada fundamentada em:

$$P_{i+1} = P_i + (1/6)(k_1 + 2k_2 + 2k_3 + k_4)h. \quad (8)$$

O cálculo de P_{i+1} é influenciado pelos parâmetros k_1 , k_2 , k_3 e k_4 , os quais são estabelecidos por:

$$\begin{aligned} k_1 &= f(t_i, P_i), \\ k_2 &= f(t_i + h/2, P_i + k_1h/2), \\ k_3 &= f\left(t_i + h/2, P_i + \frac{k_2h}{2}\right), \\ k_4 &= f(t_i + h, P_i + hk_3). \end{aligned} \quad (9)$$

3. Materiais e Métodos

Considerou-se um problema de dinâmica populacional para simular o crescimento do número de habitantes do município de Rio Grande segundo os modelos de Malthus e Verhulst, com base no número de residentes de anos anteriores, disponibilizados por órgãos governamentais. Destaca-se que durante a realização dos estudos populacionais presentes neste trabalho, não havia uma estimativa demográfica para Rio Grande no ano de 2020, sendo o ano de 2010 o último censo oficial registrado pelo Instituto Brasileiro de Geografia e Estatística (IBGE). Para trabalhar com dados mais recentes, optou-se por utilizar a base fornecida pelo DATASUS (Datasus, 2022), entre 2010 e 2012 (Tabela 1).

Tabela 1. População de Rio Grande segundo o DATASUS entre 2010 e 2012.

Ano	População
2010	197228
2011	198049
2012	198842

Fonte: DATASUS (2022)

Dada a relação ano e população, fornecida pelo DATASUS para o município de Rio Grande, utilizou-se o Jupyter Notebook com a finalidade de integrar textos explicativos e execução de códigos Python (Menezes, 2014), para estabelecer simulações temporais e estudos comparativos entre as soluções. Este arquivo está armazenado na plataforma de hospedagem de código-fonte GitHub (disponível em: <https://github.com/LaboratorioPythonFURG>) (Figura 1) para controle de versionamento e disponibilização à comunidade acadêmica.

Para efetuar as tarefas em Python, empregou-se a linguagem de programação acompanhada de suas bibliotecas NumPy (NumPy, 2023), SymPy (SymPy, 2023) e Matplotlib (Matplotlib, 2023). O NumPy é utilizado para lidar com vetores e operações matemáticas. A obtenção das soluções funcionais do problema ocorre em duas etapas: obtendo as soluções analíticas, sem a utilização de softwares, e pelo emprego da biblioteca SymPy. A solução numérica foi obtida a partir da implementação (Ruggiero, 2019) dos métodos de Euler, Euler Modificado e Runge-Kutta de 4ª ordem (Zill, 2016), realizadas em um script Python. Uma vez obtidas a solução analítica e as aproximações numéricas para o problema de crescimento populacional, foi exibida a representação gráfica da simulação temporal do sistema por meio da biblioteca Matplotlib. São apresentados estudos comparativos entre a solução exata do problema e aquelas

implementadas em Python. Esses estudos consistem em exibir o erro absoluto entre as aproximações numéricas e a resposta exata do problema. Tal análise busca evidenciar a acurácia das técnicas de solução apresentadas neste trabalho.

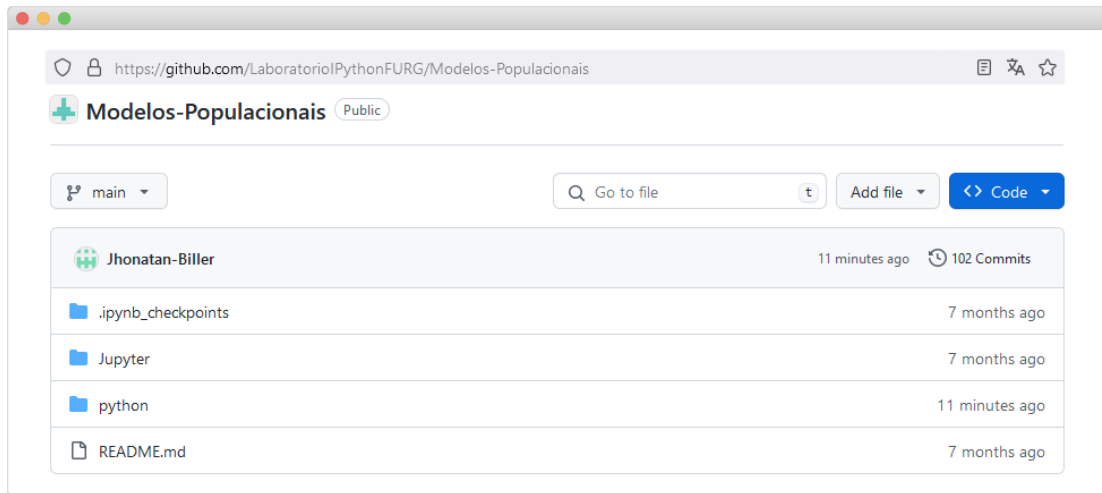


Figura 1. Repositório “Modelos-Populacionais”.

4. Resultados

Ao todo, o arquivo Jupyter, contendo os textos explicativos e as linhas de comando Python, está organizado em oito capítulos que descrevem: a proposta do problema, sua solução analítica, o emprego das bibliotecas SymPy (Figura 2) e NumPy, a implementação dos métodos numéricos de Euler (Figura 3), Euler Modificado (Figura 4) e Runge-Kutta de 4ª ordem (RK4) (Figura 5) e estudos comparativos entre os resultados obtidos (Figura 6). Cada capítulo, relacionado aos métodos de solução, contempla em suas subseções: apresentação do pseudocódigo e implementação computacional.

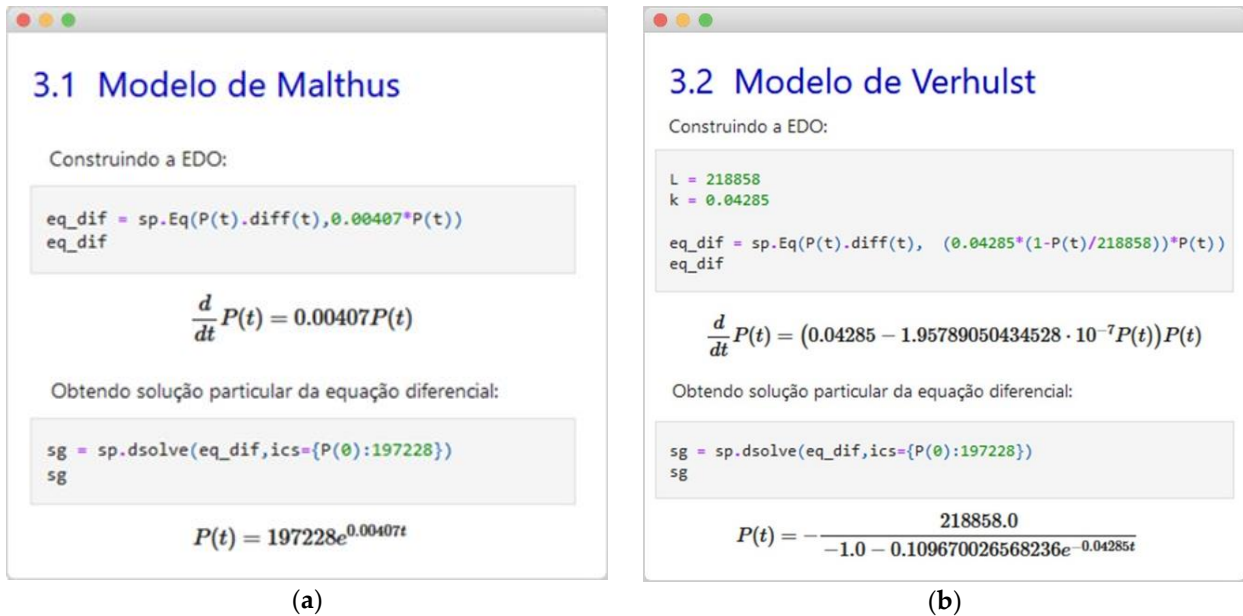


Figura 2. Arquivos Jupyter Notebook. (a) Obtenção da solução analítica do modelo de Malthus por meio da biblioteca SymPy; (b) Obtenção da solução analítica do modelo de Verhulst por meio da biblioteca SymPy.

```

1     import numpy as np #Biblioteca
2
3     def sol_euler(vt,P0,modelo): #Função Euler
4
5         P = np.array([P0]) #Vetor população
6         h = vt[2] - vt[1] #Passo de derivação
7
8         for t in vt:
9             Pi = P[-1]+h*modelo(P[-1])
10            P = np.append(P,Pi)
11
12            return P[0:P.shape[0]-1]

```

Figura 3. Implementação do Método de Euler.

```

1     import numpy as np #Biblioteca
2
3     def sol_euler_mod(vt,P0,modelo): #Função Euler Modificado
4
5         P = np.array([P0]) #Vetor população
6         h = vt[2] - vt[1] #Passo de derivação
7
8         for t in vt: #Aproximando T(i) pelo método de Euler modificado
9             Pi = P[-1]+(h/2)*(modelo(P[-1])+modelo(P[-1]+h*modelo(P[-1])))
10            P = np.append(P,Pi)
11
12            return P[0:P.shape[0]-1]

```

Figura 4. Implementação do Método de Euler Modificado.

```

1     import numpy as np
2
3     def sol_rk4(vt,P0,modelo):
4
5         P = np.array([P0])
6         h = vt[2] - vt[1]
7
8         for t in vt:
9             k1 = modelo(P[-1])
10            k2 = modelo(P[-1]+(h/2)*k1)
11            k3 = modelo(P[-1]+(h/2)*k2)
12            k4 = modelo(P[-1]+h*k3)
13            K = (1/6)*(k1+2*(k2+k3)+k4)
14            Pi = P[-1]+h*K
15            P = np.append(P,Pi)
16
17            return P[0:P.shape[0]-1]

```

Figura 5. Implementação do Método Runge-Kutta de 4ª ordem.

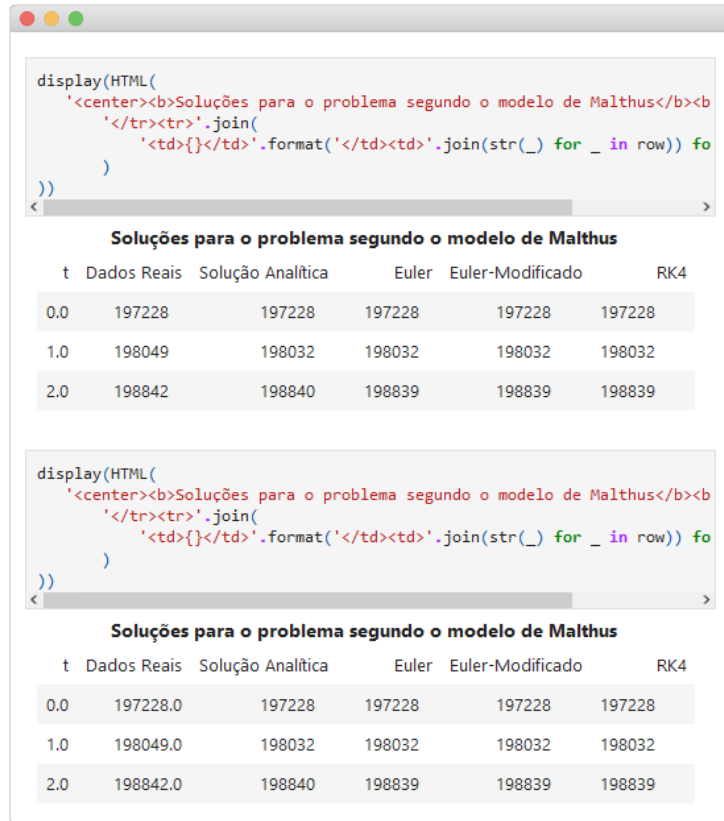


Figura 6. Exemplo de estudo comparativo.

Os arquivos Jupyter Notebook estão armazenados no repositório Modelos-Populacionais da organização LaboratórioPythonFURG, na plataforma GitHub. A organização dessa coleção de algoritmos está estruturada em dois diretórios: o primeiro contendo os arquivos Python e, o segundo, os Jupyter Notebook (.ipynb). Além disso, há um único arquivo ipynb contendo a integração dos capítulos, conforme exibe a Figura 7.

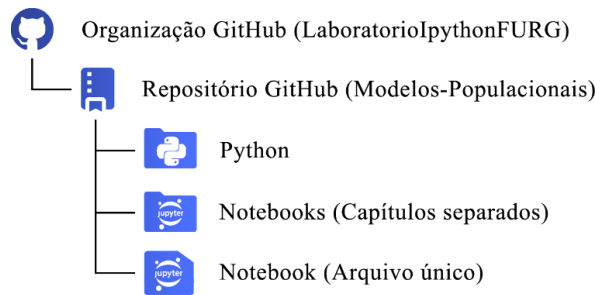


Figura 7. Estruturação do projeto.

A implementação no Jupyter Notebook das soluções e documentação foram construídas alternando entre células de código Python e HTML (Linguagem de Marcação de Hipertexto), relacionando execução de códigos Python e texto livre, conforme exemplifica a Figura 2.

Para expor graficamente os resultados obtidos através dos métodos de solução, utilizou-se o Python aliado à biblioteca Matplotlib, que permite visualizar o comportamento dos modelos de dinâmica populacional. Por exemplo, foi possível ilustrar como o crescimento populacional se relaciona com a capacidade do meio biológico para os modelos de Malthus e Verhulst pelo método RK4, como exibe a Figura 8.

Ao obter a solução numérica para os modelos populacionais, o erro percentual entre os dados reais e as aproximações para a solução foram similares para os métodos de solução abordados. O envelope de erro entre as soluções numéricas e os dados fornecidos pelo DATASUS é menor que 1%, sendo as soluções são suficientemente próximas independente do método abordado. As Tabelas 2 e 3 exemplificam esse resultado, apresentando um

comparativo entre os dados reais e as aproximações obtidas através dos métodos abordados neste trabalho para problema formulado pelo Modelo de Malthus e Verhulst, respectivamente.

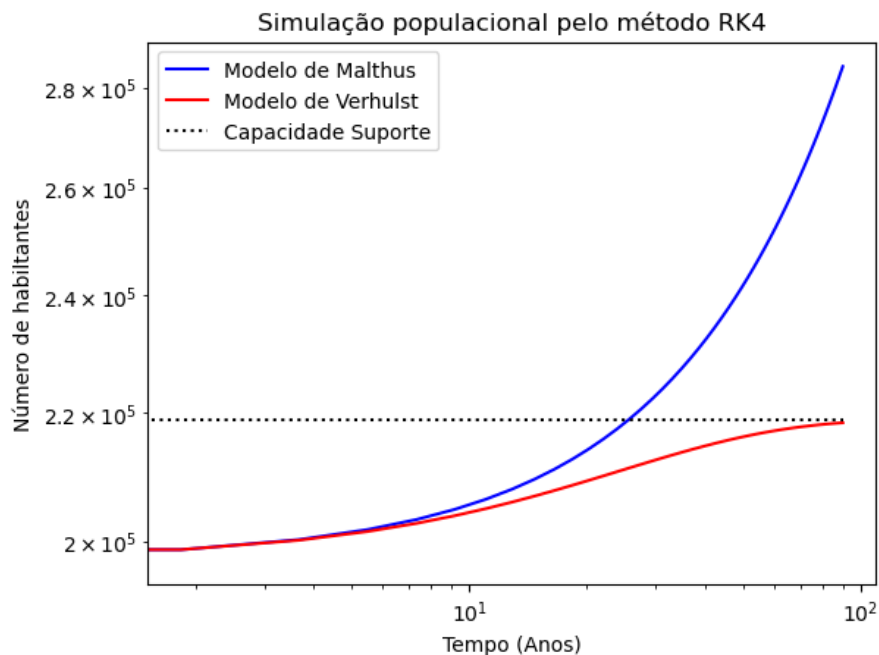


Figura 8. Simulação populacional – Método RK4.

Tabela 2. População de Rio Grande segundo o Modelo de Malthus por método de solução.

Ano	População	Analítica	Euler	Euler Modificado	RK4
2010	197228	197228	197228	197228	197228
2011	198049	198032	198032	198032	198032
2012	198842	198840	198839	198839	198839

Tabela 3. População de Rio Grande segundo o Modelo de Verhulst por método de solução.

Ano	População	Analítica	Euler	Euler Modificado	RK4
2010	197228	197228	197228	197228	197228
2011	198049	198049	198039	198049	198049
2012	198842	198842	198842	198842	198842

Além disso, para evidenciar a acurácia dos métodos numéricos em relação à solução analítica, gerou-se uma base de dados contendo o erro entre as aproximações numéricas e a solução exata dos modelos de dinâmica populacional. De forma geral, o método Runge-Kutta, que detém mais estimativas para inclinações entre dois pontos, possuem erros absolutos mais próximos de zero, conforme exemplificam as Tabelas 4 e 5.

Tabela 4. Erro absoluto entre soluções numéricas e exata para o Modelo de Malthus.

Ano	Euler	Euler Modificado	RK4
2010	0	0	0
2011	16×10^{-4}	$2,0 \times 10^{-9}$	1×10^{-10}
2012	32×10^{-4}	$4,3 \times 10^{-9}$	1×10^{-10}

Por fim, o IBGE possui uma estimativa populacional para o município de Rio Grande para o ano de 2021 (Disponível em: <https://cidades.ibge.gov.br/brasil/rs/rio-grande/panorama>), que corresponde a 212881 habitantes. Utilizou-se as

soluções analíticas dos modelos de Malthus e Verhulst para simular a população para 2021 e exibir a diferença percentual entre a estimativa modelada pelo IBGE e as estimativas dos modelos abordados neste trabalho (Tabela 6).

Tabela 5. Erro absoluto entre soluções numéricas e exata para o Modelo Verhulst.

Ano	Euler	Euler Modificado	RK4
2010	0	0	0
2011	13×10^{-3}	$1,8178 \times 10^{-7}$	$3,0 \times 10^{-11}$
2012	27×10^{-3}	$3,5312 \times 10^{-7}$	$1,5 \times 10^{-10}$

Tabela 6. Comparativo entre a estimativa do IBGE e modelos de Malthus e Verhulst para 2021.

Modelo	Simulação	Erro (%)
Malthus	206258	3,11%
Verhulst	204836	3,78%

5. Conclusões

A linguagem de programação Python, aliada a sua biblioteca NumPy, proporciona uma maneira simples e eficiente de lidar com a criação e manipulação de arranjos, seja na discretização do tempo ou construção de vetores contendo as soluções para o problemas de dinâmica populacional. Essa característica permitiu escrever os algoritmos para as técnicas de solução em poucas linhas de código. Mais ainda, considerando-se os métodos de solução advindos de bibliotecas Python, tem-se a estrutura de comandos menor e mais eficiente em termos de esforço humano. Por sua vez, o Jupyter Notebook permitiu construir a documentação do trabalho, exibindo os códigos Python, comentários e explicações sobre o problema, tabelas e figuras (biblioteca Matplotlib) em uma única página. Outra funcionalidade do Jupyter é a opção de executar uma célula de códigos por vez, tornando a manutenção das páginas Notebook mais eficiente.

Os resultados obtidos indicam que o modelo de Verhulst permite ajustar com muita precisão três anos igualmente espaçados, ao passo que o modelo de Malthus aproxima apenas dois pontos de forma satisfatória. Embora para os anos de 2010 a 2012 o modelo de Verhulst seja mais adequado, a simulação populacional para o ano de 2021 permite identificar que o modelo de Malthus melhor descreve a dinâmica da população do município de Rio Grande. Sabendo que foram variados os fatores que influenciaram o crescimento populacional da cidade, como por exemplo, a vinda de uma população para trabalhar no Polo Naval, entende-se que para o futuro novos estudos podem contribuir para o conhecimento e registro populacional do município.

Como proposta de continuidade, pretende-se estudar outras técnicas de solução implementadas na biblioteca SciPy, tais como o método RK45 e LSODA. Com a divulgação do resultado do senso demográfico para o município de Rio Grande para o ano de 2022, planeja-se realizar novas simulações considerando essa informação. Ainda, deseja-se empregar o modelo de Montroll para esses novos estudos.

Agradecimentos: Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq, através do Programa Institucional de Bolsas de Iniciação Científica – PIBIC, pelo auxílio financeiro que possibilitou a dedicação ao projeto. À Universidade Federal do Rio Grande, através do edital de Ensino, Pesquisa, Extensão e Cultura (EPEC), pelo auxílio financeiro que possibilitou dar continuidade ao projeto e a dedicação ao projeto.

Referências

1. Bacaër, N. *A Short History of Mathematical Population Dynamics*, 1a ed.; Springer: Londres, Inglaterra, 2011; Volume 1, pp. 31–39. DOI: 10.1007/978-0-85729-115-8.
2. Mill, J. S.; A Lei do Aumento da Mão-de-Obra. Em *Princípios de Economia Política: Com algumas de suas aplicações à filosofia social*, 1a ed.; Nova Cultural: São Paulo, Brasil, 1996; Volume 1, pp. 211–2017.
3. Boyce, E. W.; Richard, C.D. *Euações Diferenciais Elementares e Problemas de Valores de Contorno*, 8a ed.; LTC: Rio de Janeiro, Brasil, 2006; pp. 13–14.
4. *Python*: Página de documentação oficial do *Python*. Disponível online: <https://www.python.org/doc/> (Acesso em 30/03/2023).

5. DATASUS: Estimativas de 1992 a 2021. Disponível online: <https://datasus.saude.gov.br/populacao-residente/> (Acesso em 07/06/2022).
6. Malthus, T. R. *An Essay on the Principle of Population*; J. Johnson: London, 1798.
7. Chapra, S. C.; Canale, R. P. *Métodos Numéricos para Engenharia*, 5a ed.; Mcgraw Hill: São Paulo, 2008.
8. Menezes, N. *Introdução à Programação com Python: Algoritmos e Lógica de Programação para Iniciantes*, 2a ed.; NOVATEC: São Paulo, Brasil, 2014; Volume 1, pp. 26–29.
9. NumPy: Página de documentação do NumPy. Disponível online: <https://numpy.org/doc/stable> (Acesso em 04/06/2023).
10. SymPy: Documentação oficial do SymPy. Disponível online: <https://github.com/sympy/sympy/releases> (Acesso em 30/03/2023).
11. Matplotlib: Using Matplotlib – Matplotlib documentation. Disponível online: <https://matplotlib.org/stable/users/index.html> (Acesso em 30/03/2023).
12. Ruggiero, M.A.; *Cálculo Numérico: Aspectos teóricos e computacionais*, 2a ed.; Pearson Makron Books: São Paulo, Brasil, 2019; Volume 1, pp. 316–353.
13. Zill, D.; *Equações Diferenciais com Aplicação em Modelagem*, 10a ed.; Cengage Learning: São Pulo, Brasil, 2016; Volume 1, pp. 37–89.