Universidade Federal do Pará
Campus Universitário de Castanhal
Curso de Licenciatura em Matemática

Portfólio

O Papel da Produção Acadêmica na Construção da Identidade do Professor de Matemática

---

Renato Vinicius Costa da Silva

Castanhal
2023

O Papel da Produção Acadêmica na Construção da Identidade do Professor de Matemática

Portfólio Acadêmico, produzido como Trabalho de Conclusão de Curso, apresentado à comissão examinadora da Faculdade de Matemática do Campus de Castanhal da Universidade Federal do Pará, como requisito parcial para obtenção do grau de Licenciatura em Matemática.

Renato Vinicius Costa da Silva
renatovixk@gmail.com
Graduando em Licenciatura Plena em Matemática
UFPA-Castanhal

Banca Examinadora:

Prof. Valdelírio da Silva e Silva
Orientador

Profª. Dra. Roberta Modesto Braga
Faculdade de Matemática – UFPa/Castanhal

Prof. Dr. Renato Germano Reis Nunes
Faculdade de Matemática – UFPa/Castanhal

Castanhal
2023

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD
Sistema de Bibliotecas da Universidade Federal do Pará
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)

D111p da Silva, Renato Vinicius Costa.
O Papel da Produção Acadêmica na Construção da Identidade do Professor de Matemática / Renato Vinicius Costa da Silva. — 2023.
39 f. : il. color.

Orientador(a): Prof. Dr. Valdelírio da Silva Silva
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal do Pará, Campus Universitário de Castanhal, Faculdade de Matemática, Castanhal, 2023.

1. Formação. 2. Professor. 3. Matemática. I. Título.

CDD 510.28553

Agradecimentos

Em primeiro lugar gostaria de agradecer à minha vó, Dalvina Maria da Conceição Costa, pelo amor, carinho, sacrifício e cuidado comigo e com todos aqueles que estavam sob o seu zelo. Eu não estaria escrevendo este texto sem sua dedicação aos filhos, netos, bisnetos e agregados. Nos momentos difíceis o seu legado me deu forças, me inspirou a seguir e continuar. Levarei seus ensinamentos comigo aonde quer que eu vá.

À minha mãe, que cuidou de mim como pôde em circunstâncias adversas. Hoje a senhora é minha melhor amiga, parceira nos momentos de dificuldade e conselheira nos corriqueiros impasses da vida.

Ao meu primo Marcos Costa, que me proporcionou de diversas formas viver a universidade. Inspirou-me a refletir sobre as causas sociais, além de fomentar fincanceiramente a minha condição de estudante. Tenho imensa gratidão por tudo que fizeste por mim e prometo restribuir em breve.

Aos meus estimados amigos Clarice Neves, Afonso Barata, Luiz Vilhena, Jackeline Mesquita, Hércules Peixoto, Daniel Peixoto, Thais Silva, Hellen Cordeiro e Igor Pinheiro. Cada um de vocês contribuiu significativamente e foram decisivos para persistência durante a minha caminhada.

Às minhas primas Thuane, Hanna, Kauane Costa e também às minhas irmãs Glenda, Anayla e Anabel Reis, bem como aos meus vários outros familiares – somos muitos. Os dias são mais suportáveis com vocês por perto. Obrigado pela companhia e incentivo.

À UFPA, pelo suporte, ensino e contínuo fomento ao desenvolvimento. Agradeço pela assistência e incentivo, aos professores, coordenadores e técnicos. À direção da Faculdade de matemática, representada pela professora doutora Roberta Braga, dedico meus agradecimentos pela paciência e disposição.

Aos professores João Fumegalli, Ricardinho Ribeiro, Hugo Pena, Rafael Passinho e Leandro Matos. Cada um de vocês foi responsável pelo meu desenvolvimento como profissional, me orientando nas questões correlatas ao mercado profissional, moldando a bússula moral e intelectual do que é necessário à atividade de professor. Muito obrigado!

Por último, mas não menos importante, agradeço ao meu orientador Prof° Valdelírio Silva. Com paciência tendendo ao infinito, me ajudou a não somente confeccionar este documento, como também esteve presente em toda minha trajetória acadêmica, dando suporte, conselhos e direcionamento com alteridade, inteligência e disposição. Muitíssimo obrigado professor!

Epígrafe

"Se eu vi mais longe, foi por estar sobre ombros de gigantes."

(Sir Isaac Newton)

Resumo

Este texto constitui um trabalho de conclusão de curso no formato portfólio acadêmico, cujo objetivo é promover o diálogo entre a construção do conhecimento e a organização-reflexão a respeito do processo de aprendizado. Aqui reúnem-se coleções de documentos que corroboram na formação do autor como professor de matemática, de modo a apresentar as produções e refletir a trajetória acadêmica. Os trabalhos apresentados aqui são provenientes de estudo das componentes curriculares do curso de licenciatura plena em matemática. Dessa forma, a elaboração deles surgiu conforme a progressão natural dos conteúdos e a consequente vontade de se aplicar as ideias discutidas em sala de aula. Os dois primeiros trabalhos foram publicados em eventos realizados em Castanhal – Pa, sendo o primeiro apresentado no IV SIEPEX (Simpósio de Ensino, Pesquisa e Extensão) e o segundo no III SAMATC (Semana Acadêmica de Matemática de Castanhal). O terceiro foi submetido no XL CNMAC (Congresso Nacional de Matemática Aplicada e Computacional). Aquele versa sobre uma proposta de codificação e discussão sobre o polinômio de Taylor, uma poderosa ferramenta matemática para aproximação de funções relativamente complexas em funções polinomiais, já o segundo é um estudo e reflexão sobre o uso das técnicas de resolução de sistemas lineares de mais do que 3 equações, utilizando o Sympy – pacote da liguaguem de programação Python – para resolver os sistemas de forma simbólica, o que oferece ao aluno maior perspectiva sobre a aplicabilidade dos problemas matemáticos em contextos reais. Por fim, o último trata de uma proposta de modelagem da distribuição do calor através de uma placa 2D utilizando autômatos celulares.

Palavras-chaves: Formação; Professor; Matemática;.

Abstract

This essay is a course completion project in the form of an academic portfolio. The studies presented here come from the analysis of curricular components of the lincentiate in mathematics program. In this way, the studies were produced according to the natural progression of the content and the resulting desire to apply the ideas presented during in-class discussions. Two out of the three papers were published at events held in Castanhal – Pa. The first one was presented at the IV SIEPEX (Teaching, Research and Extension Symposium) and the second at the III SAMATC (Castanhal Mathematics Academic Week). The third one was submitted for presentation at the XL CNMAC (National Congress of Computational and applied Mathematics). The first is about a codification proposal and discussion of the Taylor polynomial, a powerful mathematical tool for approximating relatively complex functions into polynomial ones, the second is a study and reflection on the use of techniques for solving linear systems of more than 3 equations, using Sympy - a Python programming language package - to solve the systems in symbolic form, which gives the student a greater perspective on the applicability of mathematical problems in real life. Finally, the third one is a modelling proposal of heat distribution over a 2D plate using cellular automata.

Keywords: Training; Teacher; Mathematics.

Sumário

Introdução

Este documento constitui um portfólio acadêmico dos trabalhos submetidos em eventos no âmbito da licenciatura em matemática. Portanto, este portfólio visa ser uma ferramenta de reflexão e documentação que permitirá aos leitores, em particular aos alunos candidatos a professor, explorar e compreender os processos de aprendizagem, desenvolvimento pessoal e profissional. Ele se propõe a oferecer uma visão aprofundada das experiências, reflexões e ações do autor, revelando a evolução do pensamento, práticas e autoconhecimento ao longo do tempo. O portfólio é concebido como um diálogo contínuo entre o formando e o formador, destacando a importância da autorreflexão, da aprendizagem ativa e da coconstrução do conhecimento (SÁ-CHAVES, 2007). Além disso, visa fornecer uma oportunidade para analisar criticamente as práticas e promover a transformação progressiva das mesmas, resultando em um maior entendimento e aprimoramento da capacidade profissional. O portfólio também busca ser uma peça única e singular que testemunha o crescimento e o desenvolvimento do autor ao longo de sua jornada, promovendo a construção negociada da autonomia e a autoconsciência.

Aqui apresentam-se um resumo expandido, um relato de experiênica e um pôster que foram ministrados ao decorrer da graduação. Tais modadalidades são práticas de suma importância para o desenvolvimento de habilidades inerentes ao ser professor. Elas aprimoram, por exemplo, a competência da fala e capacidade de arguição, que é componente fundamental na formação de um bom docente. Ademais, a vivência no contexto de eventos em matemática propicia aprendizados e trocas de experiências que engrandecem vertiginosamente a bagagem de conhecimento do discente, tornando-se essencialmente um alicerce na contrução da identidade de um professor.

O texto foi escrito de modo a ser coletânea estruturada que reflete minha jornada acadêmica e profissional em relação à minha linha de pesquisa. É dividido em três seções interconectadas que, juntas, proporcionam uma visão abrangente da minha evolução intelectual, criativa e profissional. O primeiro tópico chamei de **Trajetória de Linha de Pesquisa**, nesta seção, apresento uma retrospectiva das minhas origens na minha linha de pesquisa, destacando os principais marcos que me levaram a escolher esse caminho. Abordo minha motivação inicial, influências intelectuais e teóricas, e descrevo como minha trajetória tem evoluído ao longo do tempo. Isso inclui minha formação acadêmica, experiências de pesquisa e eventos significativos que moldaram minha abordagem. Por outro lado, o segundo foi intitulado **Trabalhos Confeccionados**. Neles reúno uma seleção dos trabalhos e projetos mais representativos que produzi ao longo da minha jornada de pesquisa. Cada trabalho é acompanhado por uma breve descrição, contexto e resumo. Essa seção destina-se a ilustrar minha contribuição concreta para a linha de pesquisa, destacando minha capacidade de pesquisa, análise e produção acadêmica ou profissional. Por fim, tem-se as **Reflexões Finais**, a última seção do portfólio que é dedicada a reflexões finais sobre a minha trajetória de pesquisa e os trabalhos apresentados. Aqui, busco contextualizar o que aprendi, desafios enfrentados e o impacto dessas experiências na minha formação e prática. Além disso, discuto minhas ambições futuras na linha de pesquisa, áreas de aprimoramento e o que espero alcançar no próximo estágio da minha jornada acadêmica ou profissional.

1 Trajetória da linha de pesquisa

A universidade é mais do que uma instituição de ensino superior; é um epicentro de conhecimento, uma forja para o crescimento intelectual e uma incubadora de ideias que moldam o futuro. É um lugar onde o desejo de aprender, a curiosidade e a ambição se encontram para criar uma experiência transformadora. Em um curso de matemática, essa jornada é particularmente intrigante, especialmente quando combinada com a crescente importância da programação e da matemática aplicada.

Antes de mergulharmos nas especificidades da trajetória de quatro anos de estudos que me dediquei, é importante destacar o impacto social da universidade. A educação universitária não é apenas um caminho para o desenvolvimento pessoal; é uma força motriz para o progresso da sociedade. A universidade capacita indivíduos a se tornarem agentes de mudança, a desafiar o *status quo* e a desenvolver soluções para os problemas complexos do mundo. Em um contexto global cada vez mais interconectado, a pesquisa e a educação universitária desempenham um papel crucial na construção de um futuro mais promissor.

Ao iniciar o curso de matemática, a jornada que fiz foi marcada pela exploração de conceitos fundamentais, desde cálculo e álgebra até geometria e probabilidade. Os primeiros anos foram como a base de uma pirâmide, fornecendo o conhecimento essencial necessário para construir estruturas conceituais mais complexas nos anos subsequentes. A matemática pura é a linguagem da lógica e da abstração, e nesse estágio, desenvolvi habilidades analíticas e pensamento crítico que são valiosos em qualquer área.

No decorrer dos estudos, a programação surgiu como uma ferramenta poderosa. A matemática e a programação se demonstraram parceiras naturais, pois ambas envolvem a resolução de problemas, a manipulação de dados e a criação de algoritmos. A programação torna os matemáticos capazes de aplicar seu conhecimento em cenários do mundo real. Aprender a codificar permite que os estudantes automatizem tarefas, realizem análises de dados avançadas e desenvolvam modelos matemáticos interativos. Através da programação, a matemática deixa de ser uma disciplina abstrata e ganha vida no contexto da tecnologia e da ciência de dados, desenvolvi boa parte das minhas habilidades à luz dessas ideias.

Nos anos intermediários, a ênfase se volta para a matemática aplicada. Como alunos, começamos a compreender como os princípios matemáticos são usados em setores como engenharia, economia, ciências naturais e até mesmo nas ciências sociais. A matemática aplicada é a ponte que conecta a teoria matemática com a resolução de problemas reais. Ela fornece as ferramentas necessárias para modelar fenômenos do mundo real, desde a propagação de doenças até o comportamento do mercado de ações. Foi nessa fase que percebi a importância prática da educação em matemática.

Governado por ideias que pudessem ser aplicadas conforme ensinado através dos anos, decidi investir tempo e energia em trabalhos e habilidades que me capacitacem nas áreas dos trabalhos publicados. O primeiro, na estrutura deste texto, trata-se de uma discussão singela, porém consistente, das ideias do matemático Brook Taylor, famoso pelas Séries de Taylor, utilizando a linguagem de programação *Python*, imprescindível para o avanço da matemática computacional. Sob o título de **O polinômio de Taylor e sua implementação em código *Python*: uma breve discussão**, este foi resultado de estudo da disciplina de cálculo 1 e também de iniciação à programação, componentes curriculares obrigatórias da Faculdade de Matemática da UFPA - Campus Castanhal. Tal trabalho foi apresentado no evento supracitado como comunicação oral.

O segundo trabalho, submetido como *relato de experiência*, nasceu das discussões junto ao Prof° Valdelírio Silva. Com o título de **O *Sympy* como proposta no ensino de sistemas lineares aplicados**, foi escrito sob o ponto de vista educacional, incentivado pela incongruência entre a matemática ensinada em sala de aula e os problemas mais realistas. Nesse sentido, observamos que conforme a complexidade dos problemas aumentavam, havia um escalonamento na quantidade de cálculos a serem realizados, sendo, portanto, necessária a utilização de ferramentas que pudessem agilizar tais processos. Assim, deu-se origem ao texto, que foi apresentado na forma oral, assim como primeiro. Ambos estão disponíveis nos anexos no final deste documento.

O terceiro, apresentado no formato de pôster, versa sobre uma alternativa ao uso de métodos tradicionais para modelagem de problemas em matemática. Em particular, nele é apresentada a ideia de âutomato celular com sua respectiva aplicação na distribuição do calor em uma placa retangular. Sob o título de **Difusão de Calor em uma Placa Retangular Usando Autômatos Celulares** fez-se uma análise e emprego do método para se obter o comportamento e portanto prever a temperatura em pontos da placa. Utilizando linguagem *R*, foi mostrado que, comparativamente, o algoritmo é suficientemente vantajoso na demanda de custo computacional. Além disso, valeu-se de gráficos para mostrar a difusão e a convergência dos resultados.

Optei, portanto, no modelo de portfólio como modalidade de Trabalho de Conclusão

Curso, que não apenas serve como uma ferramenta de avaliação do desempenho acadêmico, mas também o de papel de estratégia para incentivar a aprendizagem baseada em competências, enfatizando a abordagem 'aprender fazendo' (SÁ-CHAVES, 2007). Isso ocorre devido às experiências adquiridas ao longo do percurso acadêmico, que são concebidas com o propósito de enriquecer a formação educacional. Com base nesses princípios escolhi criar um portfólio que inclui os trabalhos submetidos em eventos os quais participei nos anos de 2021 e 2023.

2

Os trabalhos confeccionados

1. **O polinômio de Taylor e sua implementação em código *python*: uma breve discussão**

   Autoria: Renato Vinicius Costa da Silva & Arthur da Costa Almeida

   Ano: 2021

   Categoria: Submetido resumo expandido com apresentação por comunicação oral.

   Evento: IV SIEPEX – Simpósio de Ensino, Pesquisa e Extensão do campus Castanhal – UFPA;

   Resumo: Este trabalho buscou apresentar o Polinômio de Taylor, sua implementação em *Python* e alguns exemplos-aplicações dele por meio de algorítmos, uma vez que muitos comportamentos da natureza são descritos por funções matemáticas comumente não triviais, que frequentemente requerem manipulações algébricas complicadas ou computacionalmente despendiosas. Nesse sentido, parece natural procurar por funções mais simples, como as que envolvem produtos e somas, isto é, polinômiais. Diante disso, o polinômio de Taylor surge como uma ferramenta poderosa para aproximar resultados daquelas funções. Além disso, o uso de recursos computacionais automatizam os procedimentos para obter a aproximação de Taylor. Por isso, usaremos a linguagem de programação de alto nível *Python*, por ter uma sintaxe concisa e clara, junto com sua vasta biblioteca de módulos científicos, é uma das mais indicadas para nosso objetivo.

2. **O *Sympy* como proposta no ensino de sistemas lineares aplicados.**

   Autoria: Renato Vinicius Costa da Silva & Valdelírio da Silva e Silva

   Ano: 2023

   Categoria: Submetido resumo expandido com apresentação por comunicação oral em *relato de experiência.*

Evento: III SAMATC – Semana Acadêmica de Matemática de Castanhal

Resumo: Mediante a necessidade cada vez maior de se fazer uso de tecnologias computacionais no ensino, softwares de matemática simbólica tornam-se mais importantes. Este trabalho apresenta aplicações matemáticas de sistemas lineares resolvidos pela bibioteca *Sympy* do *Python* com uma aplicação expansão de sistema de ordem menores discutidos comumente no ensino desse assunto; dando explicitamente a sintaxe para obtenção da solução, assim como de comandos para sua visualização gráfica. Os códigos não estão simplesmente para utilização do professor, mas para orientação na criação e discussão conjunta com os discentes, configurando-se como parte do letramento em lógica de programação

3. **Simulação de Difusão de Calor em uma Placa Retangular Usando Autômatos Celulares.**

Autoria: Renato Vinicius Costa da Silva & Arthur da Costa Almeida

Ano: 2021

Categoria: Submetido resumo com apresentação de pôster virtual.

Evento: XL Congresso Nacional de Matemática Aplicada e Computacional - CNMAC 2021

Resumo: Este texto discute o estudo da difusão de calor em uma placa usando modelos computacionais baseados em autômatos celulares. A equação do calor é apresentada, juntamente com a definição de autômatos celulares como uma grade regular de células com regras de transição locais. A pesquisa compara dois tipos de vizinhança para autômatos celulares, Neumann e Moore, com o objetivo de modelar a condução de calor em uma placa. Duas simulações são realizadas, uma usando a média das células da vizinhança de Moore e outra com um esquema derivado de diferenças finitas com vizinhança de Neumann. Os resultados das simulações são comparados com o valor exato obtido pela solução analítica. A temperatura da célula central após as simulações é de 44,34°C e 44,29°C, enquanto o valor exato é 44,42°C. Gráficos mostram a distribuição de calor na placa e a convergência para a solução exata ao longo das iterações.

2.1 O polinômio de Taylor e sua implementação em código *python*: uma breve discussão

Em 2021, este trabalho se concentrou na exploração do Polinômio de Taylor, um conceito apresentado em Cálculo 1, e sua implementação em *Python*. O objetivo principal foi compreender como esse polinômio pode ser utilizado para aproximar funções matemáticas relativamente complexas.

A motivação para esse estudo reside na necessidade de lidar com funções matemáticas que, muitas vezes, são difíceis de analisar diretamente. Essas funções podem envolver manipulações matemáticas complicadas ou requerer muitos recursos computacionais para sua avaliação. Nesse contexto, os polinômios se destacam como uma abordagem mais simples e acessível, já que são compostos por somas e produtos de termos.

Para realizar esse estudo, utilizamos a linguagem de programação *Python* devido à sua clareza e facilidade de uso. Além disso, a vasta biblioteca de módulos científicos disponíveis no *Python* facilitaram a implementação e a análise de várias funções matemáticas complexas.

O projeto incluiu a criação de algoritmos que automatizaram a aplicação do Polinômio de Taylor, tornando o processo mais eficiente e preciso. O uso do *Python* simplificou a implementação desses algoritmos e proporcionou a flexibilidade necessária para explorar e aplicar o Polinômio de Taylor em diversos cenários.

2.2 O *Sympy* como proposta no ensino de sistemas lineares aplicados

Em 2023, diante da crescente necessidade de incorporar tecnologias computacionais ao ensino, os softwares de matemática simbólica assumem um papel de destaque. Este trabalho surgiu como uma resposta à demanda de explorar problemas matemáticos mais alinhados com a realidade, abraçando a eficácia das ferramentas digitais em ambientes educacionais.

A principal ênfase deste projeto recai sobre a aplicação de conceitos matemáticos relacionados a sistemas lineares. Para atingir esse objetivo, utilizamos a biblioteca *Sympy* do *Python*, uma poderosa ferramenta de matemática simbólica. Nosso enfoque concentrou-se em discutir e resolver sistemas lineares desde ordem 2 até ordem 4, expandindo um tópico comumente abordado no ensino dessa disciplina.

Uma característica fundamental deste trabalho é a apresentação da sintaxe necessária para obter soluções para esses sistemas, bem como a demonstração de comandos que possibilitam sua visualização gráfica. Isso não apenas simplifica o processo de resolução, mas também torna o entendimento dos resultados mais acessível para alunos e professores.

Um aspecto particularmente relevante deste projeto é que os códigos gerados não se destinam apenas ao uso dos professores, mas também à orientação e discussão conjunta com os alunos. Isso se alinha com a ideia de promover o letramento em lógica de programação, capacitando os estudantes a compreender e aplicar conceitos matemáticos de forma prática, uma habilidade valiosa na era da tecnologia.

2.3 Simulação de Difusão de Calor em uma Placa Retangular Usando Autômatos Celulares.

No contexto da pesquisa, a difusão de calor desempenha um papel fundamental em inúmeras situações do mundo real. Por exemplo, em engenharia, compreender como o calor se propaga em estruturas é crucial para o projeto de sistemas de refrigeração eficientes, prevenção de superaquecimento em equipamentos eletrônicos e otimização do isolamento térmico em edifícios.

Além disso, o estudo da difusão de calor é essencial na pesquisa de processos naturais, como a transferência de calor na atmosfera, nos oceanos e na Terra, e influencia a modelagem climática e ambiental. Essa compreensão é crucial para lidar com questões ambientais, como mudanças climáticas e eventos climáticos extremos, onde a difusão de calor desempenha um papel central na distribuição de temperaturas.

No entanto, a modelagem precisa da difusão de calor muitas vezes envolve equações diferenciais parciais complexas, que podem ser computacionalmente intensivas e desafiadoras de resolver. É nesse contexto que a pesquisa apresentada no texto se destaca, propondo a aplicação de autômatos celulares como uma abordagem alternativa e eficaz para estudar esse fenômeno.

A capacidade dos autômatos celulares de considerar o estado das células vizinhas e aplicar regras locais torna-os uma ferramenta promissora na análise da difusão de calor, simplificando o processo de modelagem e fornecendo resultados que se aproximam das soluções analíticas. Essa abordagem inovadora tem o potencial de revolucionar a forma como abordamos a compreensão e simulação da difusão de calor em diversas aplicações práticas e científicas.

3

Reflexões finais: a condição de professor

Em síntese, percebo a construção da identidade professor de matemática através das produções acadêmicas. De fato, existem habilidades, valores e competências que são paulatinamente formadas mediante a terna pesquisa-elaboração-apresentação das produções. Além, claro, de todo contexto ambiental dos eventos, logística e interação com os colegas, que deságuam também na contrução das habilidades, valores e competênicas.

Durante os anos de graduação que me prepararam para essa jornada, pude adquirir um arcabouço de conhecimento teórico e prático que agora carrego como minha principal ferramenta. Aprender sobre pedagogia, psicologia educacional, métodos de ensino e tantos outros aspectos da educação foi enriquecedor. Mas, mais do que isso, foi a experiência de estar em sala de aula, de interagir com os alunos, de aprender com seus questionamentos e descobertas que moldou minha visão de professor.

Cada desafio enfrentado, cada aula ministrada e cada conquista dos alunos reforçaram a convicção de que a docência é uma das profissões mais significativas que alguém pode abraçar. Ser professor não se trata apenas de transmitir conhecimento, mas de inspirar, guiar, e, acima de tudo, acreditar no potencial de cada estudante.

Nossa responsabilidade vai além das notas e dos resultados. Estamos moldando mentes, nutrindo sonhos e preparando os jovens para um futuro que, muitas vezes, nem nós mesmos podemos prever completamente. Essa é uma missão que exige dedicação, paciência e um comprometimento inabalável.

À medida que concluo este texto, vejo um mundo em constante evolução, onde as demandas sobre a educação são cada vez mais complexas e diversificadas. É por isso que a minha jornada como professor está longe de ser um ponto final; na verdade, é apenas o começo de uma busca contínua por aprimoramento e adaptação. Além disso, cada sala de aula será um novo desafio, e cada aluno será uma nova oportunidade de aprender e crescer. Como professor, estou comprometido em enfrentar esses desafios de cabeça erguida, com humildade e empatia. Pois, afinal, o maior aprendizado que levei desta jornada de preparação é que, como professor, sou também um eterno aprendiz.

Conforme encerro este portfólio, sinto-me grato pela oportunidade de trilhar este caminho e inspirado pelo que o futuro reserva. Sei que as lições aprendidas e os princípios que guiam minha prática como professor me capacitarão a impactar positivamente as vidas daqueles que passarem por minha sala de aula, assim como eu fui afetado positivamente pelos meus professores.

Referências

ALBAGLI, S. Divulgação científica: informação científica para cidadania. *Ciência da informação*, v. 25, n. 3, 1996.

AZEVEDO, I. B. d. *O prazer da produção científica; diretrizes para a elaboração de trabalhos acadêmicos. 10ª*. Piracicaba, SP: São Paulo: Hagnos, 2001.

FAZENDA, I. C. A. *A pesquisa em educação e as transformações do conhecimento*. Campinas, SP: Papirus Editora, 1995.

GARRIDO, E. et al. A pesquisa colaborativa universidade/escola, a formação do professor reflexivo/investigativo e a construção coletiva de saberes e práticas pela equipe escolar. projeto usp-ayres. *Anais. Olhando a qualidade do ensino a partir da sala de aula. Resumos*, 1998.

MIRANDA, M. d. O professor pesquisador e sua pretensão de resolver a relação entre a teoria e a prática na formação de professores. *O papel da pesquisa na formação e na prática dos professores*, v. 5, p. 129–143, 2001.

PAIVA, M. Saberes profissionais de professores que ensinam matemática: um diálogo com professores experientes. *XII Seminário de Investigação em Educação Matemática*, p. 209–232, 2001.

PIMENTA, S. G.; GHEDIN, E. Professor reflexivo no brasil: gênese e crítica de um conceito. 2008.

SÁ-CHAVES, I. *Portfólios Reflexivos–Estratégias de Formação e de Supervisão/Reflective Portfolios–Training and Supervisory Strategies*. Aveiro : Universidade: Formação de Professores, Cadernos Didácticos–Série Supervisão, 2007. 8, 12

Apêndice

O POLINÔMIO DE TAYLOR E SUA IMPLEMENTAÇÃO EM CÓDIGO PYTHON : UMA BREVE DISCUSSÃO

Renato Vinícius Costa da Silva
Universidade Federal do Pará
renatovixk@gmail.com

Arthur da Costa Almeida
Universidade Federal do Pará
arthur@ufpa.br

Resumo:

Este trabalho buscou apresentar o Polinômio de Taylor, sua implementação em Python e alguns exemplos-aplicações dele por meio de algorítmos, uma vez que muitos comportamentos da natureza são descritos por funções matemáticas comumente não triviais, que frequentemente requerem manipulações algébricas complicadas ou computacionalmente despendiosas. Nesse sentido, parece natural procurar por funções mais simples, como as que envolvem produtos e somas, isto é, polinômiais. Diante disso, o polinômio de Taylor surge como uma ferramenta poderosa para aproximar resultados daquelas funções. Além disso, o uso de recursos computacionais automatizam os procedimentos para obter a aproximação de Taylor. Por isso, usaremos a linguagem de programação de alto nível Python, por ter uma sintaxe concisa e clara, junto com sua vasta biblioteca de módulos científicos, é uma das mais indicadas para nosso objetivo.
Palavras-chave: Python, Polinômio de Taylor, recursos computacionais, programação.

Introdução

Quando descrevemos certos problemas de natureza matemática, é comum que eles não possuam soluções simples ou elas são algebricamente cansativas, sendo necessário outras saídas. Uma delas é através de argumentos numéricos, ou seja, aproximações. Desse modo, o polinômio de Taylor é vantajoso para aproximar f(x) = y, pois nele trabalhamos apenas com somas, produtos e derivação, o que é bem útil quando desejamos empregar algorítmos computacionais. Ademais, foi por sua demonstração pioneira de que é possível aproximar uma função derivável em torno de um ponto x0 mediante a cálculos que dependiam das derivadas da função nesse ponto que o Polinômio de Taylor recebeu o nome do matemático britânico Brook Taylor (PEIXOTO, 2019). Com isso, o problema que outrora era desanimador, agora, sob certas condições, é relativamente simples. Por outro lado, apesar de servir para enfrentar determinadas questões, dependendo do grau do polinômio, as contas vão se intensificando, o que pode ser exaustivo, ou seria, caso não usássemos alternativas computacionais. Nesse contexto, é interessante a inclusão de algorítmos para automatizar a tarefa de efetuar os cálculos no caminho de encontrar a aproximação. Escreveremos, então, o código em Python, por diversos motivos mais adiante discutidos. Neste, usaremos ideias básicas de programaçao estrutural, não ultilizando a parte de orientação a objeto, pois é sufiente a parte inicial. Por fim, vamos mostrar o quão úteis podem ser as ideias de Taylor munidas do poder computacional atual, à guisa de Guido Van Rossum, Criador do Python.

Metodologia

Este trabalho será feito por meio de levantamento bibliográfico, explorando as ideias de Taylor, junto com a ferramenta computacional Python.

O polinômio

De acordo com Anton, Bivens & Davis (2007), Se f puder ser derivada n vezes em x_0 , então definiremos o enésimo polinômio de Taylor de f em torno de $x = x_0$ como sendo

$$p(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \cdots + \frac{f^n(x_0)}{n!}(x - x_0)^n$$

Ou ainda, mais abreviadamente,

$$p(x) = \sum_{k=0}^{n} \frac{f^k(x_0)}{k!}(x - x_0)^k$$

No caso em que $x_0 = 0$, consoante Anton, Bivens & Davis (2007), os polinômios de Taylor são chamados de polinômios de Maclaurin. Assim, todos os teoremas do primeiro também se aplicam ao segundo. Note ainda que só faz sentido calcular o polinômio se tivermos algumas garantias, como a existência de n derivadas, bem como o cálculo destas no ponto x_0. Nisso, não é difícil ver que conforme nos distanciamos deste, maior será a diferença entre o valor de $f(x)$ e $p(x)$. Por isso, se faz necessário um controle dessa diferença, para saber qual é o erro que cometemos. No entanto, iremos suprimir esses conceitos, dado caráter introdutório deste artigo.

A linguagem Python

O Python é uma linguagem de programação de alto nível que possibilita aos seus usuários trabalhar com uma sintaxe concisa e clara, recursos poderosos da sua biblioteca padrão e módulos e frameworks desenvolvidos por terceiros. Sua linguagem projetada com a filosofia de enfatizar a importância do esforço do programador sobre o esforço computacional, prioriza a legibilidade do código sobre a velocidade ou expressividade ao que inclui diversas estruturas de alto nível (listas ,tuplas, dicionários, data/hora, complexos e outras) (PEIXOTO, 2019). Além disso, existem bibliotecas que podem ser importadas direto dos ambientes de desenvolvimento integrado (IDE's) para fins mais específicos, como é o caso do SymPy, uma poderosa biblioteca que usaremos para matemática simbólica, por ser um sistema de ágebra computacional completo, enquanto mantém o script simples e compreensível. Por fim, usaremos também o PyCharm, um IDE performático, dinâmico e de fácil navegação, especificamente construído para a linguagem Python e cujo mantenedor é a empresa checa JetBrains.

O algorítmo

O algorítmo foi escrito com ênfase no resultado. Considerando a sintaxe, houve linhas apenas para adequação às ideias matemáticas pertinentes ao polinômio de Taylor. Isso posto, temos

Código 1: O algorítmo que calcula o polinômio de Taylor nas proximidades de x_0

```
"""
Aproximação de Taylor
"""
from sympy import *
from math import e
x = symbols('x')

funcao = sympify(input('Informe a função: ')) # recebe a função já em Sympy
x0 = N(input('Na vizinhança de qual numero? ')) # N() reconhece numeros reais ,
    incluido pi , por exemplo
grau = int(input('Ate o polinomio de que grau? '))
```

```python
px = 0
derivada = funcao # Apesar de ser logicamente estranho, essa parte ajuda no código
if 'e' in str(funcao): # Para o algorítmo entender o numero de Euler
    funcao = funcao.subs('e', e)
if 'pi' in str(funcao): # Para o algoritmo entender o numero pi
    funcao = funcao.sub('pi', pi)
for i in range(1, grau+1):
    for k in range(i):
        if i == 1:
            px = funcao.subs(x, x0) # Calcula f em x0
        derivada = diff(derivada, x) # Auxilia no somatório
    px += (derivada.subs(x, x0) / factorial(i)) * (sympify('x') - x0)**i # Fórmula para calcular o polinomio
    derivada = funcao
print(f'O polinomio de Taylor associado a funcao fornecida é \n {px}')
```

Comparações

Vamos agora apresentar resultados obtidos por meio do código e compara-los aos valores encontrados por um computador. Também vamos plotar os gráficos dos polinômios, assim como os das funções que lhes dão origem. Para tanto, usaremos o software gratuíto Geogebra, disponível para download em Geogebra, uma programa eficiente, intuitivo e dinâmico, ideal para os nossos objetivos. Nesse sentido, temos A solução x pelo código é retornada como sendo

$$f(x) = tg(3x) + x^3 - 7x, \text{ próximo a } x = \frac{2}{5}$$
$$\text{Valor na calculadora em } x_0 = \frac{3}{7} \text{ é } 0.48$$
$$\text{Valor encontrado pelo polinômio de grau 3 no mesmo } x_0 \text{ é } 0.49$$

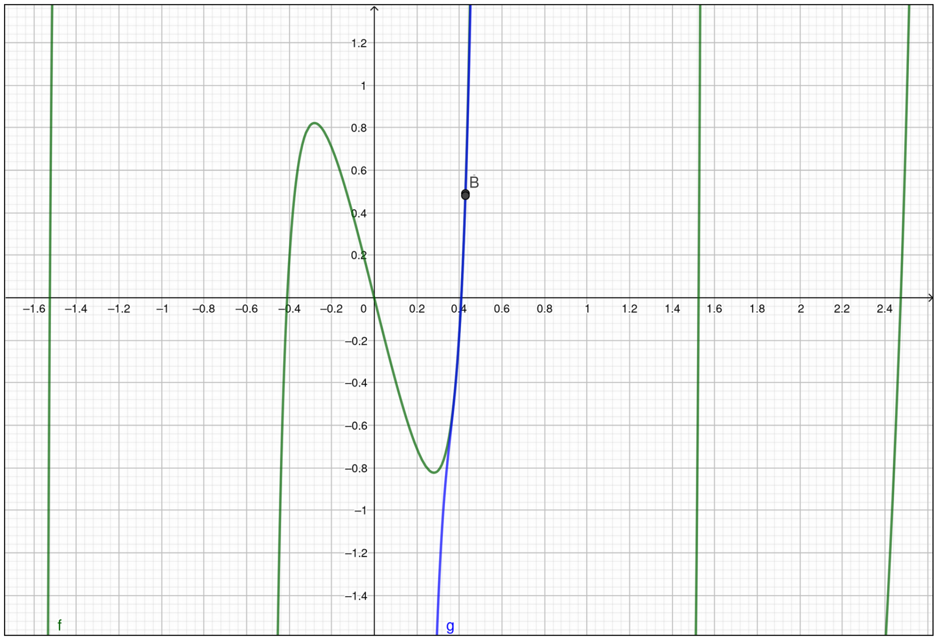
Observe o gráfico em 1, nele, em verde, temos função f e em azul o polinômio $g(x)$ encontado pelo algorítmo, a saber $g(x) = 16.3278919016212x + 1429.99114163477(x - 0.4)^3 + 177.504726650765(x - 0.4)^2$˘$6.69500513852215$. O ponto A possui abcissa em x_0 e ordenada em $f(x_0)$, também temos o ponto B com mesma abcissa e ordenada $g(x_0)$. Note a aproximação de A e B com desenvolvimento de um polinômio com grau relativamente baixo!

$$f(x) = tg(e^{3x} - \pi) + cos^3(5x) - \pi x, \text{ próximo a } x = -0.5$$
$$\text{valor na calculadora em } x_0 = -0.6 \text{ é } 1.08$$
$$\text{valor encontrado pelo polinômio de grau 6 no mesmo } x_0 \text{ é } 1.12$$

Geometricamente, perceba o gráfico em 2, nele, em lilás, temos função f e em cinza o polinômio $g(x)$ encontado pelo algorítmo, a saber $g(x) = -1353.61020104568(x+0.5)^6 + 1498.42921144109(x+0.5)^5 + 169.082296125582(x+0.5)^4 - 139.759496686589(x+0.5)^3 - 1.07570452767624(x+0.5)^2 + (5.76177045329511)(x+0.5) - 0.287290461930553$

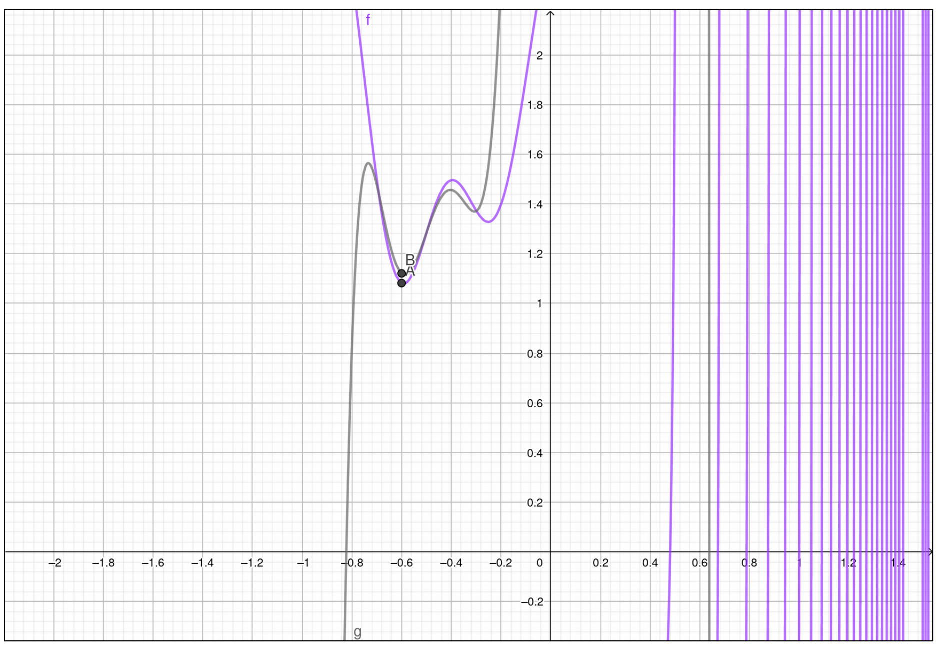
O ponto A possui abcissa em x_0 e ordenada em $f(x_0)$, também temos o ponto B com mesma abcissa e ordenada $g(x_0)$. Note a adquação do gráfico do poliômio ao da função próximo a $x = -0.5$, é de fato uma boa aproximação. O ponto A, comparado a B, ilustra bem o poder que a ferramenta de Taylor nos fornece.

Figura 1: Comparando f e g em x_0



Fonte: do autor

Figura 2: Comparação de f e g próximos ao ponto x_0



Fonte: do autor

Considerações finais

As discussões das ideias de Taylor foram imprescíndiveis para o avanço tecnológico da sociedade. Ainda hoje, explorar o polinômio é uma vantagem para quem busca modelar e/ou calcular funções mais delicadas. Logarítmo, por exemplo, não é munido de um procedimento que resolva todos os casos, uma forma genérica de encontrar qualquer solução (vale a pena pensar sobre como calcular log(1.003)). Este trabalho apresentou o Polinômio, sua implementação em código Python e alguns resultados consequentes desses. Mostramos como a aplicação deles torna, relativamente, mais simples pensar em certos objetos matemáticos, e visamos desenvolver tudo isso de forma a despertar o interesse dos leitores no assunto, assim como sobre programação e sobre os recursos tecnológicos disponíveis hoje. A matemática e a computação, juntas, são motores que fomentam a civilização no sentido do progresso e da harmonia, com elas nos tornamos cidadãos mais conscientes da realidade em que estamos e podemos exercer nossas obrigações e reivindicar nossos direitos mais plenamente.

Referências

ANTON, H.; BIVENS, I. C.; DAVIS, S. L. *Cálculo-Volume I-8*. [S.l.]: Bookman, 2007.

COELHO, F. C. *Computação Científica com Python*. [S.l.]: Lulu. com, 2007.

FRANCO, N. B. *Cálculo numérico*. [S.l.]: Pearson, 2006.

PEIXOTO, H. H. d. S. *Contrução e implementação de algorítmos escritos em código Python para cálculos de interpolação*. 58 f. Monografia (Graduação) — Faculdade de Licenciatura em Matemática, Universidade Federal do Pará, Castanhal, 2019.

CERTIFICADO

Certificamos que o trabalho intitulado **O POLINÔMIO DE TAYLOR E SUA IMPLEMENTAÇÃO EM CÓDIGO PYTHON : UMA BREVE DISCUSSÃO** foi apresentado, em forma de comunicação, durante o **IV SIMPÓSIO DE ENSINO, PESQUISA E EXTENSÃO** realizado, *online*, no período de 25 a 27 de agosto de 2021 pelo *campus* de Castanhal da Universidade Federal do Pará – UFPA.

AUTORES: Renato Vinicius Costa da Silva; Arthur da Costa Almeida.

FRANCISCO VALDINEI SANTOS DOS ANJOS
COORDENADOR ACADÊMICO UFPA- CASTANHAL

FACOMP NAcess GEPLISES UFPA

ISSN: 2595 7856

III SAMATC
25 a 27 outubro 2023
SAMATC
SEMANA ACADÊMICA DE MATEMÁTICA DE CASTANHAL
Universidade Federal do Pará - Campus Castanhal
MATEMÁTICA, CIÊNCIA E TECNOLOGIA: FORMAÇÃO DOCENTE E APLICAÇÕES NO CONTEXTO AMAZÔNICO.

O *Sympy* como Proposta no Ensino de Sistemas Lineares Aplicados

Renato Vinícius Costa da Silva
Universidade Federal do Pará
renatovixk@gmail.com

João Carlos Correa Amador
SEDUC – Pa

Valdelírio da Silva e Silva
Universidade Federal do Pará
valdel@ufpa.br

Resumo:

Mediante a necessidade cada vez maior de se fazer uso de tecnologias computacionais no ensino, softwares de matemática simbólica tornam-se mais importantes. Este trabalho apresenta aplicações matemáticas de sistemas lineares resolvidos pela bibioteca *Sympy* do *Python* com uma aplicação expansão de sistema de ordem menores discutidos comumente no ensino desse assunto; dando explicitamente a sintaxe para obtenção da solução, assim como de comandos para sua visualização gráfica. Os códigos não estão simplesmente para utilização do professor, mas para orientação na criação e discussão conjunta com os discentes, configurando-se como parte do letramento em lógica de programação.
Palavras-chave: Sistemas Lineares. Matemática Simbólica. *Sympy*. TIDCs.

Introdução

A matemática é uma linguagem universal que permeia nossas vidas de maneiras incontáveis. No entanto, o ensino tradicional muitas vezes nos limita a uma visão simplificada desse vasto universo matemático. Neste trabalho, propomos romper um pouco essas limitações, sugerindo uma exploração mais abrangente e prática da matemática, especificamente no que diz respeito aos sistemas lineares, utilizando o potencial da computação de matemática simbólica *Sympy* do *Python*.

Muitos de nós lembramos das aulas de matemática no ensino fundamental, onde os sistemas lineares eram frequentemente apresentados em sua forma mais simples, com poucas incógnitas e cálculos manuais que não refletiam a complexidade da matemática do mundo real. A pergunta que levantamos aqui é: Por que não explorar sistemas lineares mais complexos que vão além da ordem 3? A resposta é simples: para evitar que a matemática se torne uma série de exercícios didaticamente dispendiosos e impraticáveis em uma sala de aula. Logo, vamos explorar o potencial de usar programação para trabalhar com sistemas lineares em problemas reais, adicionando um toque de contexto e aplicação prática à matemática.

À medida que nos aventuramos em sistemas lineares de ordens mais elevadas, nos deparamos com desafios computacionais significativos. A complexidade aumenta significativamente em relação aos de ordem 2 e 3, especialmente quando envolvemos cálculos de determinantes, cujo número de elementos cresce com o fatorial da ordem da matriz (POOLE, 2017)(Work with Symbolic Matrices). Esta complexidade não é apenas um problema teórico, mas também uma questão prática quando ensinamos aos alunos a realizar manipulações simbólicas e operações em matrizes. Veremos como a escalabilidade afeta nossas abordagens de ensino e como a matemática simbólica do *Python* pode ser uma aliada nesse cenário. Desse modo, O *SymPy*, uma biblioteca *Python* de matemática simbólica (Sympy Documentation), nos permite estender as possibilidades de resolução de sistemas lineares. Após ensinar aos alunos os fundamentos da resolução de sistemas de ordem baixa (de preferência, por nós autores, por eliminação gaussiana!), é conveniente explorar como o *SymPy* pode ser empregado para abordar problemas do mundo real que envolvem sistemas de equações de ordens superiores. Esta abordagem não apenas torna a matemática mais aplicada, mas também oferece a oportunidade de contextualizar conceitos matemáticos, tornando o aprendizado mais envolvente e prático. Portanto, nossa proposta é ensinar os alunos a realizar a eliminação gaussiana, não apenas com números, mas também com operações matemáticas simbólicas fazendo uso da sintaxe do *SymPy*; em seguida, mostrar as funções do próprio *Sympy* para resolver problemas matemáticos envolvendo sistemas lineares, embora que sejam relativamente simples, demandam tempo e operações extensas quando resolvidos manualmente.

Resolução Simbólica de Sistemas com o *SymPy*

A primeira etapa que sugerimos acima para o ensino de resolução de sistemas lineares não se apresenta aqui. Em vez disso, dedicamos colocar neste texto discussões de sistemas aplicados à própria matemática, com soluções obtidas pelos próprios recursos do *Sympy* (ARI; MAMATNAZAROVA, 2014; MEURER et al., 2017) e visualizadas pela utilização do *Matplotlib* (Matplotlib Documentation).

Sistema linear de duas incógnitas

A fim de ilustar a sintaxe do *SymPy* para solução de um sistema linear consideremos o problema de determinar a equação de uma reta r no $\mathbb{R}^2$ que passa por dois pontos conhecidos, (x_1, y_1) e (x_2, y_2). Sabemos que na forma reduzida a reta r se apresenta por

$$y = mx + h,$$

e o que estamos interessados são justamente determinar os coeficientes m e h. Dessa forma, com os dois pontos conhecidos temos as duas equações

$$x_1 \cdot m + 1 \cdot h = y_1$$

e

$$x_2 \cdot m + 1 \cdot h = y_2,$$

que matricialmente tem a configuração:

$$\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \end{bmatrix} \cdot \begin{bmatrix} m \\ h \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

Nesse contexto, para o *SymPy* devemos usar as entradas do sistema como matrizes A e b para encontrar a matriz solução $x = [m\ h]^t$. A sintaxe é dada no código 1 a seguir:

Código 1: Solução do sistema de determinação de coeficiente angular e linear da reta

```
from sympy import symbols
from sympy.matrices import Matrix
x_1, y_1, x_2, y_2 = symbols("x_1, y_1, x_2, y_2")
A = Matrix([[x_1,1],[x_2,1]])
b = Matrix([y_1,y_2])
x = A.solve(b)
```

A solução x pelo código é retornada como sendo

$$x = \begin{bmatrix} \dfrac{y_1 - y_2}{x_1 - x_2} \\ \\ \dfrac{x_1 y_2 - x_2 y_1}{x_1 - x_2} \end{bmatrix}$$

Para o exemplo de terem sido dados os pontos $(2,3)$ e $(-1,4)$ basta fazer as substituições das coordenadas na expressão da solução obtida. Isso é conseguido com o comando *subs*, cuja codificação para tais pontos, que resultou em $m = \frac{-1}{3}$ e $h = \frac{11}{3}$, foi a seguinte:

Código 2: Solução numérica por substituição particular na solução

```
m = x[0].subs([(x_1,2),(y_1,3),(x_2,-1),(y_2,4)]) #coeficiente angular
h = x[1].subs([(x_1,2),(y_1,3),(x_2,-1),(y_2,4)]) #coeficiente linear
```

Apesar do *SymPy* poder plotar gráficos de funções dadas implicitamente, vamos utilizar apenas o pacote *Matplotlib* pois é o mais conhecido com uso do Python[1]. Com ele no entanto é necessário apresentar a equação da forma implícita com membro direito nulo. Para este exemplo deve-se entrar considerando $y + \frac{1}{3}x - \frac{11}{3} = 0$, isso porque para a curva ser mostrada, primeiro deve-se criar um grid de pontos no domínio x e contra-domínio y e sobre ele tomar os pontos que satisfazem a equação nula. Pelo *SymPy* bastaria usar o comando *plot_implicit*, mas infelizmente nele não é tão simples inserir pontos isolados como os que são dos dados informados para o sistema linear! Decorrente disso foram confeccionados códigos para plotagem de gráficos de funções implícitas bidimensionais e tridimensionais. O primeiro é uma subrotina (ou seja, *function*) que possibilita entrar com um função implícita que aceita os limites das bordas a serem considerados no gráfico, junto com a entrada das coordenadas de dois pontos. Essa *function* é o código 3 abaixo:

Código 3: Subrotina para plotagem da reta que passa pelos pontos dados.

```
def plota_implicita2pontos(f, bordas, P1, P2):
    from numpy import linspace, meshgrid
    xmin = bordas[0]
    xmax = bordas[1]
    ymin = bordas[2]
    ymax = bordas[3]
    fig = plt.figure()
    ax = fig.add_subplot(111)
```

[1] O *Matplotlib* é o pacote que “roda” por trás dos recursos gráficos do *SymPy*

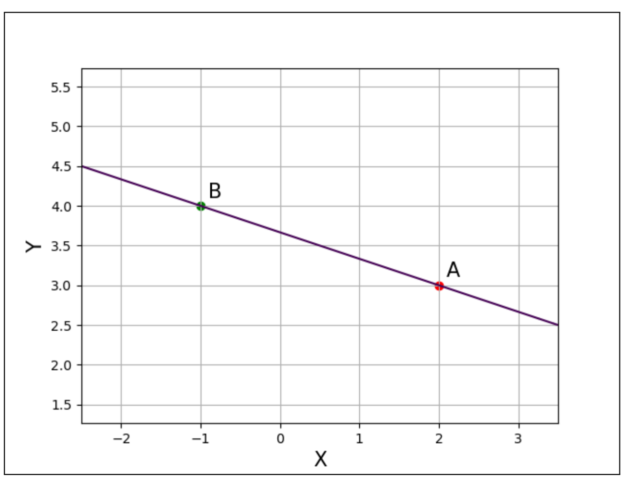
```
    A = linspace(xmin, xmax, 100)  # discretização do eixo X
    B = linspace(ymin, ymax, 100)  # discretização do eixo Y
    X, Y = meshgrid(A, B)          # grid do domínio XoY
    Z = f(X, Y)
    ax.contour(X, Y, Z, [0])
    ax.scatter(P1[0], P1[1], color='red', alpha = 1)
    ax.scatter(P2[0], P2[1], color='green', alpha = 1)
    ax1 = fig.gca()
    ax1.text(P1[0]+0.1, P1[1]+0.1, 'A', fontsize = 15)
    ax1.text(P2[0]+0.1, P2[1]+0.1, 'B', fontsize = 15)
    ax.set_xlim(xmin, xmax)
    ax.set_ylim(ymin, ymax)
    ax.set_aspect('equal', 'datalim')
    ax.set_xlabel("X", fontsize = 15)
    ax.set_ylabel("Y", fontsize = 15)
    ax.grid()
    plt.show()
```

Para uma reta, o professor/aluno pode confeccionar também uma *function* com os coeficientes angular e linear dentro dela. Em seguida, informadas as bordas do gráfico e dois pontos para serem representados, a subrotina para plotagem da reta pode ser chamada como ilustrada no código 4 abaixo, exemplificado para os pontos considerados acima. Resultante desses comandos obtém-se a figura 1:

Código 4: Exemplificação de criação de subrotina de uma reta e a chamada para plotagem

```
def reta(x,y):
    m, h = -1/3, 11/3
    return y-m*x-h
bordas = [-2.5, 3.5, 1.5, 5.5]
P1 = [2, 3]
P2 = [-1,4]
plota_implicita2pontos(reta, bordas, P1, P2)
```

Figura 1: Reta obtida com a resolução do sistema linear com dois pontos dados



Fonte: do autor

Sistema linear de três incógnitas

Agora consideremos o problema de determinar a equação da circunferência que contém três pontos dados, $(x_1, y_1), (x_2, y_2)$ e (x_3, y_3). Sabemos que a equação canônica, ou padrão de uma circunferência é dada por

$$(x - x_0)^2 + (y - y_0)^2 = r^2,$$

em que (x_0, y_0) são as coordenadas do centro da circunferência e r o seu raio. Desenvolvendo os quadrados obtemos

$$x^2 - 2x_0x + x_0^2 + y^2 - 2y_0y + y_0^2 = r^2 \Leftrightarrow 2x_0x + 2y_0y + r^2 - x_0^2 - y_0^2 = x^2 + y^2,$$

ou mais simplificadamente sendo

$$ax + by + c = x^2 + y^2,$$

em que $a = 2x_0, b = 2y_0$ e $c = r^2 - x_0^2 - y_0^2$.

Sendo $(x_1, y_1), (x_2, y_2)$ e (x_3,y_3) pontos não colineares no plano, então existe uma circunferência que os contém (LIMA, 2015). Sendo assim, substituindo as coordenadas desses pontos na equação acima, teremos:

$$\begin{cases} ax_1 + by_1 + c = x_1^2 + y_1^2 \\ ax_2 + by_2 + c = x_2^2 + y_2^2 \\ ax_3 + by_3 + c = x_3^2 + y_3^2 \end{cases}$$

tendo a, b e c as incógnitas que desejamos determinar, as quais depois de encontradas, podem ser substituídas em suas definições para encontrarmos o centro (x_0, y_0) e o raio r da circunferência. O código para resolução via *SymPy* é o 5 seguinte:

Código 5: Determinação da equação da circunferência dados três pontos

```
from math import sqrt
from sympy import symbols
from sympy.matrices import Matrix
x_1, y_1, x_2, y_2, x_3, y_3 = symbols("x_1, y_1, x_2, y_2, x_3, y_3")
A = Matrix([[x_1,y_1,1],[x_2,y_2,1],[x_3,y_3,1]])
B = Matrix([x_1**2+y_1**2,x_2**2+y_2**2,x_3**2+y_3**2])
x = A.solve(B)
```

Teremos então $x_0 = \dfrac{a}{2}, y_0 = \dfrac{b}{2}$ e $r = \sqrt{c + x_0^2 + y_0^2}$, as quais são encontradas pelo uso do comando *subs*, codificadas a seguir para o exemplo com os pontos $A(2,6), B(2,0)$ e $C(5,3)$:

Código 6: Determinação do centro da circunferência e raio pela substituição de variáveis

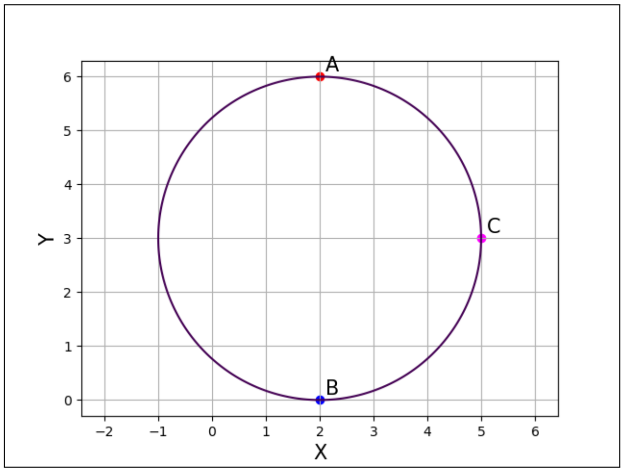
```
a = x[0].subs([(x_1,2),(y_1,6),(x_2,2),(y_2,0),(x_3,5),(y_3,3)])
x0 = a / 2
b = x[1].subs([(x_1,2),(y_1,6),(x_2,2),(y_2,0),(x_3,5),(y_3,3)])
y0 = b / 2
c = x[2].subs([(x_1,2),(y_1,6),(x_2,2),(y_2,0),(x_3,5),(y_3,3)])
r = sqrt(c + x0**2 + y0**2)
```

Para utilizar o *Matplotlib* no caso da circunferência, sua equação deve ser tomada como argumento de plotagem na forma implícita $(x - x_0)^2 + (y - y_0)^2 - r^2$. Foi construída uma subrotina para qualquer função dada implicitamente bidimensional e que sejam inseridos no gráfico três pontos. Nela os argumentos de entrada são a função implícita, um array (do tipo lista) dos limites das bordas do gráfico e as listas com as coordenadas dos três pontos. Também se tem apresentada no código 7 abaixo uma *function* que apresenta a expressão do círculo com os coeficientes deste exemplo; e também uma ilustração da chamada da subrotina que faz a plotagem. A figura (2) apresenta os resultados do problema abordado no exemplo.

Código 7: Subrotina e comandos para plotagem da circunferência com 3 pontos dados.

```python
def plota_implicita3pontos(f, bordas, P1, P2, P3):
    import matplotlib.pyplot as plt
    from numpy import linspace, meshgrid
    xmin = bordas[0]
    xmax = bordas[1]
    ymin = bordas[2]
    ymax = bordas[3]
    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax1 = fig.gca()
    A = linspace(xmin, xmax, 100)  # discretização do eixo X
    B = linspace(ymin, ymax, 100)  # discretização do eixo Y
    X, Y = meshgrid(A, B)          # grid do domínio XoY
    Z = f(X, Y)
    ax.contour(X, Y, Z, [0])
    ax.scatter(P1[0], P1[1], color='red', alpha = 1)
    ax.scatter(P2[0], P2[1], color='blue', alpha = 1)
    ax.scatter(P3[0], P3[1], color='magenta', alpha = 1)
    ax1.text(P1[0]+0.1, P1[1]+0.1, 'A', fontsize = 15)
    ax1.text(P2[0]+0.1, P2[1]+0.1, 'B', fontsize = 15)
    ax1.text(P3[0]+0.1, P3[1]+0.1, 'C', fontsize = 15)
    ax.set_xlim(xmin, xmax)
    ax.set_ylim(ymin, ymax)
    ax.set_aspect('equal', 'datalim')
    ax.set_xlabel("X", fontsize = 15)
    ax.set_ylabel("Y", fontsize = 15)
    ax.grid()
    plt.show()
# ------------------------------------------------------------
def circulo(x,y):
    x0, y0, r = 2, 3, 3
    return (x - x0)**2 + (y - y0)**2 - r**2
# ------------------------------------------------------------
x0, y0, r = 2, 3, 3
bordas = [x0-1.1*r, x0+1.1*r, y0-1.1*r, y0+1.1*r]
P1 = [2, 6]
P2 = [2, 0]
P3 = [5, 3]
plota_implicita3pontos(circulo, bordas, P1, P2, P3)
```

Figura 2: Circunferência obtida com a resolução do sistema linear com 3 pontos não colineares



Fonte: do autor

Sistema linear de quatro incógnitas

De quatro pontos não coplanares pode-se ter uma esfera que os contenha (LIMA, 2015). Dessa forma, considerando a equação geral de um esfera sendo

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = R^2,$$

em que (x_0, y_0, z_0) são as coordenadas do centro da esfera e R o raio; tem-se desenvolvendo os quadrados a expressão:

$$x^2 - 2x_0x + x_0^2 + y^2 - 2y_0y + y_0^2 + z^2 - 2z_0z + z_0^2 = R^2 \iff ax + by + cz + d = x^2 + y^2 + z^2,$$

fazendo-se as substituições $a = 2x_0, b = 2y_0, c = 2z_0$ e $d = R^2 - x_0^2 - y_0^2 - z_0^2$. Daí, tendo os pontos $(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3)$ e (x_4, y_4, z_4) monta-se o sistema linear

$$\begin{cases} ax_1 + by_1 + cz_1 + d = x_1^2 + y_1^2 + z_1^2 \\ ax_2 + by_2 + cz_2 + d = x_2^2 + y_2^2 + z_2^2 \\ ax_3 + by_3 + cz_3 + d = x_3^2 + y_3^2 + z_3^2 \end{cases}$$

sendo a, b, c e d as incógnitas.

Decorrente de aqui se ter 12 variáveis simbólicas a solução obtida tem um tempo bem maior que os dos outros exemplos. O código para esta aplicação é o código 8 a seguir, que junto também tem a substituição dos pontos $A(0, 3, 2), B(1, -1, 1), C(2, 1, 0)$ e $D(5, 1, 3)$, para obtenção do centro, retornado por $(x_0, y_0, z_0) = (2, 1, 3)$ e raio $R = 3$.

Código 8: Código para solução do sistema 4×4 com determinação de centro e raio de esfera.

```
from sympy import symbols
from sympy.matrices import Matrix
from math import sqrt
x_1, y_1, z_1, x_2, y_2, z_2 = symbols("x_1, y_1, z_1, x_2, y_2, z_2")
x_3, y_3, z_3, x_4, y_4, z_4 = symbols("x_3, y_3, z_3, x_4, y_4, z_4")
A = Matrix([[x_1,y_1,z_1,1],[x_2,y_2,z_2,1],[x_3,y_3,z_3,1],[x_4,y_4,z_4,1]])
```

```
7  B = Matrix([[x_1**2+y_1**2+z_1**2], [x_2**2+y_2**2+z_2**2],
8             [x_3**2+y_3**2+z_3**2],[x_4**2+y_4**2+z_4**2]])
9  s = A.solve(B)
10 #substituição dos pontos para cálculo dos coeficientes x0, y0, z0 e R
11 x0 = s[0].subs([(x_1,0),(y_1,3),(z_1,2),(x_2,1),(y_2,-1),(z_2,1),
12                 (x_3,2),(y_3,1),(z_3,0),(x_4,5),(y_4,1),(z_4,3)]) / 2
13 y0 = s[1].subs([(x_1,0),(y_1,3),(z_1,2),(x_2,1),(y_2,-1),(z_2,1),
14                 (x_3,2),(y_3,1),(z_3,0),(x_4,5),(y_4,1),(z_4,3)]) / 2
15 z0 = s[2].subs([(x_1,0),(y_1,3),(z_1,2),(x_2,1),(y_2,-1),(z_2,1),
16                 (x_3,2),(y_3,1),(z_3,0),(x_4,5),(y_4,1),(z_4,3)]) / 2
17 d = s[3].subs([(x_1,0),(y_1,3),(z_1,2),(x_2,1),(y_2,-1),(z_2,1),
18                (x_3,2),(y_3,1),(z_3,0),(x_4,5),(y_4,1),(z_4,3)])
19 # raio da esfera
20 R  = sqrt(d + x0**2 + y0**2 + z0**2)
```

O *SymPy* possui uma ferramenta para plotagem de gráfico de expressão dada implicitamente, mas somente para funções de duas variáveis. Para contornar, parcialmente, esse problema aqui se criou uma subrotina para esboçar o gráfico da esfera como uma superfície. O comando *surface* realiza pelo *Matplotlib* a plotagem da superfície de uma função expressa na forma $z = f(x, y)$. E no caso aqui se tem $z = \pm\sqrt{R^2 - (x - x_0)^2 - (y - y_0)^2} + z_0$, que deve ser separada em duas, uma com sinal positivo e outra com negativo. E como o procedimento de plotagem gráfica parte da discretização do domínio do plano XOY, vários valores x e y nos intervalos desse domínio não são criados, e por esse motivo não são interpolados de modo que a superfície se mostre contínua. A *function plota_esfera* é o código para visualização gráfica da esfera, que nas linhas finais do código 9 transcrito abaixo, chama a função para a consideração dos pontos A, B, C e D usados acima. A figura 3 apresenta a esfera da solução obtida pelo sistema, e nela vê-se, além de duas superfícies, descontinuidade na proximidade da circunferência horizontal de raio R. Esse problema deve ser explanado aos alunos, e a justificativa é de que não existem pontos criados entre dois números consecutivos das variáveis A e B do código (linhas 15 e 16), e não interpolados pelas duas cascas esféricas[2].

Código 9: Subrotina de plotagem da esfera e os comandos para uso do exemplo considerado.

```
1  def plota_esfera(centro, R, bordas, P1, P2, P3, P4):
2      from numpy import linspace, meshgrid, sqrt
3      import matplotlib.pyplot as plt
4      x0 = centro[0]
5      y0 = centro[1]
6      z0 = centro[2]
7      xmin = bordas[0]
8      xmax = bordas[1]
9      ymin = bordas[2]
10     ymax = bordas[3]
11     zmin = bordas[4]
12     zmax = bordas[5]
13     fig = plt.figure()
14     ax = fig.add_subplot(111, projection='3d')
```

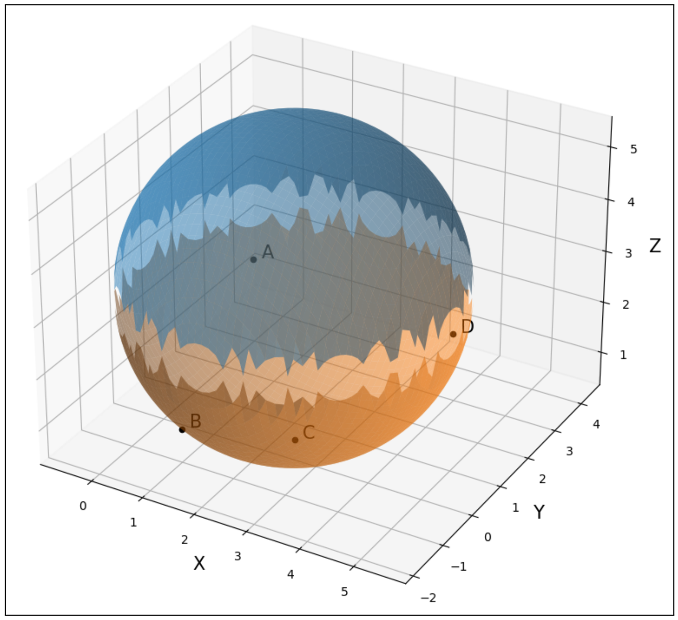
[2]Atente-se que um volume $A \times B \times Z$ deveria ser criado a partir de todos os pontos de $A \times B$, mas Z depende de raiz quadrada na sua expressão, e logicamente os pontos do volume fora da esfera não terão raiz quadrada! Por isso ao rodar o código aparecerão avisos de valores inválidos.

```
    A = linspace(xmin, xmax, 100)   # discretização do eixo X
    B = linspace(ymin, ymax, 100)   # discretização do eixo Y
    X, Y = meshgrid(A, B)          # grid de XOY
    Zp = sqrt(R**2 - (X-x0)**2 - (Y-y0)**2) + z0  #casca superior da esfera
    Zn = -sqrt(R**2 - (X-x0)**2 - (Y-y0)**2) + z0 #casca inferior da esfera
    ax.plot_surface(X, Y, Zp, alpha = 0.5)
    ax.plot_surface(X, Y, Zn, alpha = 0.5)
    ax.scatter(P1[0], P1[1], P1[2], color='black', alpha=1)
    ax.scatter(P2[0], P2[1], P2[2], color='black', alpha=1)
    ax.scatter(P3[0], P3[1], P3[2], color='black', alpha=1)
    ax.scatter(P4[0], P4[1], P4[2], color='black', alpha=1)
    ax1 = fig.gca()
    ax1.text(P1[0]+0.1, P1[1]+0.1, P1[2], 'A', fontsize = 15)
    ax1.text(P2[0]+0.1, P2[1]+0.1, P2[2], 'B', fontsize = 15)
    ax1.text(P3[0]+0.1, P3[1]+0.1, P3[2], 'C', fontsize = 15)
    ax1.text(P4[0]+0.1, P4[1]+0.1, P4[2], 'D', fontsize = 15)
    ax.set_xlim3d(xmin, xmax)
    ax.set_ylim3d(ymin, ymax)
    ax.set_zlim3d(zmin, zmax)
    ax.set_aspect('equal','datalim')
    ax.set_xlabel("X", fontsize = 15)
    ax.set_ylabel("Y", fontsize = 15)
    ax.set_zlabel("Z", fontsize = 15)
    plt.show()
#------------------------------------------------------------
# coordenadas do centro e raio:
x0, y0, z0, R = 2, 1, 3, 3
# limites para apresentação gráfica:
xi = x0 - 1.1*R
xf = x0 + 1.1*R
yi = y0 - 1.1*R
yf = y0 + 1.1*R
zi = z0 - 1.1*R
zf = z0 + 1.1*R
# inputs da function plota_esfera:
centro = [x0, y0, z0]
bordas = [xi, xf, yi, yf, zi, zf]
P1, P2, P3, P4 = [0, 3, 2], [1,-1,1], [2,1,0], [5,1,3]
#------------------------------------------------------------
# chamada para plotagem:
plota_esfera(centro, R, bordas, P1, P2, P3, P4)
```

Considerações finais

Verificamos que o uso do *SymPy* mostrou-se adequadamente útil para resolução de sistemas em termos da matemática simbólica. Ilustramos sua aplicabilidade a partir de problemas tradicionalmente abordados no ensino básico e no ensino superior, fomentando o desenvolvimento de habilidades no uso de ferramentos computacionais e, ainda que de forma tímida, extrapolamos o exemplo eclesiástico de sistema 3×3, apresentando uma discussão de um sistema de ordem maior. A importância

Figura 3: Esfera obtida como solução do sistema linear com 4 pontos não coplanares



Fonte: do autor

deste trabalho reside na busca por maior aproximação da prática em matemática aplicada da matemática discutida em sala de aula. Esta, muitas vezes, se manifesta por meios desconexos da realidade do aluno e não traduzem a complexidade necessária para se alcançar soluções de problemas factuais. Os resultados obtidos foram categóricos. Permeou-se ideiais chaves dentro do estudo das matrizes, da geometria analítica e programação em *Python*, como queríamos. Em particular, construímos arcabouço teórico suficiente para uma discussão cautelosa sobre a codificação da resolução de sistemas e sua visualização geometrica, recurso indispensável no âmbito do aprendizado. Por outro lado, é importante salientar que podem existir limitantes na diligência deste trabalho, cujo principal potencial representante é o acesso de alunos à dispositivos que sejam tecnologicamente capazes de suportar os programas apresentados. Nossa proposta está aquém dessa discusão, que pode ser alvo de investigações futuras. Por fim, destaca-se a necessidade de incorporar cada vez mais instrumentos digitais na procura pela capacitação dos discentes, sob a pena de distanciamento deles dos recursos tecnológicos atuais.

Referências

ARI, N.; MAMATNAZAROVA, N. Symbolic python. In: IEEE. *2014 11th International Conference on Electronics, Computer and Computation (ICECCO)*. Abuja, Nigeria, 2014. p. 1–8.

LIMA, E. L. *Geometria analítica e álgebra linear*. Rio de Janeiro: Instituto Nacional de Matemática Pura e Aplicada, 2015.

MEURER, A. et al. Sympy. PeerJ Inc., 2017.

POOLE, D. *Álgebra Linear: uma introdução moderna*. São Paulo: Cengage Learning, 2017.

RESULTADO DA AVALIAÇÃO

O trabalho intitulado "O Sympy COMO PROPOSTA NO ENSINO DE SISTEMAS LINEARES APLICADOS" foi **APROVADO** no evento III SAMATC – SEMANA ACADÊMICA DE MATEMÁTICA DE CASTANHAL

- **Título:** O Sympy COMO PROPOSTA NO ENSINO DE SISTEMAS LINEARES APLICADOS
- **Número:** 744231
- **Data de Submissão:** 15/10/2023
- **Modalidade:** Apresentação Oral (15 minutos) - TEMPLATE ATUALIZADO (17/10/2023).
- **Área Temática:** Educação Matemática
- **Autores:** Renato Vinícius Costa Da Silva,Valdelirio da Silva e Silva

Cordialmente,
Comissão Científica
Roberta Modesto Braga
iiisamatc@gmail.com

Simulação de Difusão de Calor em uma Placa Retangular Usando Autômatos Celulares

Renato V. C. da Silva **Laize C. C. de Souza** **Clarice das N. Barbosa**
Arthur da C. Almeida
UFPA - Universidade Federal do Pará, Faculdade de Matemática
68746-360, Campus Castanhal, PA
E-mail: renatovixk@gmail.com , laizecrist@gmail.com, clariceneves60@gmail.com, arthur@ufpa.br

RESUMO

A difusão do calor em uma placa é estudada, em geral, usando-se a solução analítica da equação do calor (1) ou métodos numéricos como o método das diferenças finitas. Neste trabalho, entretanto, a proposta é estudar um modelo computacional para esse fenômeno, baseado em autômatos celulares.

$$\frac{\partial u}{\partial t}=c^2\cdot\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right) \tag{1}$$

Um autômato celular é uma grade regular de células, geralmente quadradas, onde cada uma delas pode estar em um número finito de estados em cada unidade de tempo, junto com uma regra ou função de transição, que define em cada instante como deve ser alterado o estado de cada célula da grade [2]. A mudança de estado das células ocorre de forma sincronizada, em paralelo. Além disso, a regra é de natureza local, envolvendo principalmente o estado das células próximas, ou vizinhança. Dois tipos mais comuns de vizinhança são usados, a vizinhança de Neumann, que envolve apenas as 4 celulas localizadas ao N, S, L, O e a vizinhança de Moore, que além dessas 4 envolve também as 4 células das diagonais [3].

Segundo [4] autômatos celulares podem ser usados como uma ferramenta apropriada para o estudo de vários fenômenos físicos, tais como a condução de calor em uma placa, em lugar das tradicionais equações diferenciais parciais, conforme será feito neste trabalho.

Será estudada a condução de calor em uma placa retangular, com o lado superior mantido em uma temperatura de 100 ºC e o resto da placa, inicialmente com 0 ºC. Para a primeira simulação dessa placa foi definida uma grade no formato de uma matriz com 100 linhas e 200 colunas, inicialmente com 0 em todas as células, com exceção da primeira linha mantida no valor de 100 em todos os momentos. Para definir a regra de transição, foi considerado que cada célula assume a média das 8 células da vizinhança de Moore, mas assumindo-se que as células das diagonais, contribuem com 1/4 de seu valor por terem um contato menor com a célula central. Assim, a regra de transição de estado para cada célula $c_{(i,j)}(t+1)$ é dada por (2)

$$c_{(i,j)}(t+1) = (4*(c_{(i-1,j)}(t) + c_{(i+1,j)}(t) + c_{(i,j-1)}(t) + c_{(i,j+1)}(t)) + c_{(i-1,j-1)}(t) + c_{(i-1,j+1)}(t) + c_{(i+1,j-1)}(t) + c_{(i+1,j+1)}(t)) / 20 \tag{2}$$

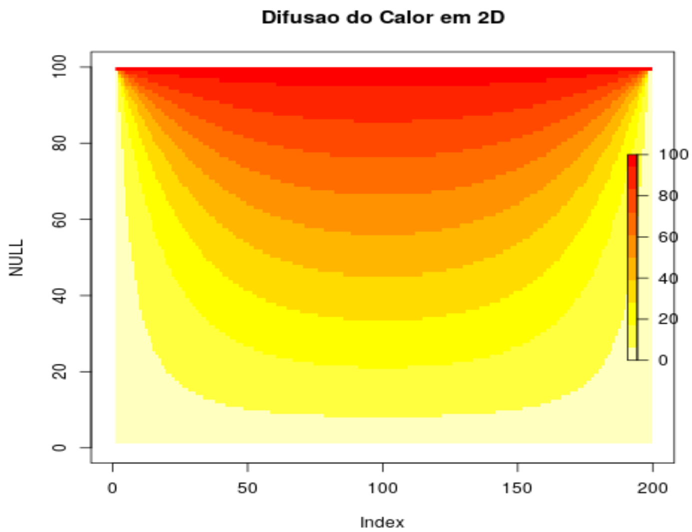
A segunda simulação foi feita com a mesma placa retangular, de 100 x 200 células, nas mesmas condições iniciais, mas agora, usando como regra de transição de estado o esquema derivado da solução numérica da equação em diferenças finitas, com vizinhança de Neumann, ficando assim (3)

$$c_{(i,j)}(t+1) = c_{(i,j)}(t) + \alpha(c_{(i,j-1)}(t) + c_{(i-1,j)}(t) -4c_{(i,j)}(t) + c_{(i+1,j)}(t)) + c_{(i,j+1)}(t)) \tag{3}$$

Como resultado da primeira simulação, usando-se o valor médio das células da vizinhança de Moore, obtém-se para a temperatura da célula central da grade o valor de 44,34 ºC, com 18 mil interações. Usando-se o esquema do método das diferenças finitas, com vizinhança de Neumann, obtém-se o valor de 44,29 ºC, com 20 mil iterações. O valor exato, dado pela solução analítica é de 44,42 ºC.

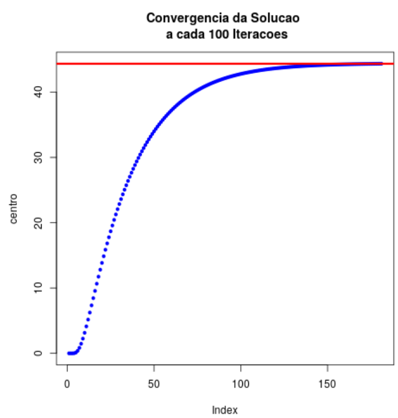
Na Figura 1 vemos o mapa da distribuição de calor na placa, na situação estacionária e na Figura 2, temos um gráfico mostrando a convergência para a solução exata (reta horizontal vermelha) a cada 100 iterações.

Figura 1. Mapa da difusão do calor



Fonte: os autores (2019)

Figura 2. Convergência da solução para o valor da solução analítica



Fonte: os autores (2019).

Palavras-chave: *autômatos celulares, difusão de calor, simulação*

Referências

[1] S. J. Farlow. "Differential Equations for Scientists and Engineers". Dover Editions, NY, 1993.

[2] J. L. Schiff. "Cellular Automata: A Discrete View of the World". Ed.Wiley-Interscience New Jersey, 2008.

[3] D. P. Feldman. "Chaos and Fractals. An Elementary Introduction". Ed Oxford University Press, UK, 2012.

CERTIFICADO

CNMAC 2021
XL Congresso Nacional de Matemática Aplicada e Computacional

Certificamos que o trabalho intitulado

Difusão de Calor em uma Placa Retangular Usando Autômatos Celulares

Autoria de

Renato Silva, Laize Souza, Arthur da Costa Almeida

foi apresentado por **Renato Silva**

como pôster durante o XL Congresso Nacional de Matemática Aplicada e Computacional - CNMAC 2021, realizado de 13 a 17 de setembro de 2021 em formato virtual.

Pablo Martín Rodríguez
Presidente
Sociedade Brasileira de Matemática Aplicada e Computacional

Organização:

SBMAC

UFMS
FUNDAÇÃO UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL