INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SUL-RIO-GRANDENSE - CÂMPUS PASSO FUNDO
CURSO BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

RICARDO AUGUSTO MÜLLER GERMANI

CONVOLUÇÃO INCREMENTAL: Um algoritmo eficiente para a computação de convoluções referentes à funções de equações diferenciais

PASSO FUNDO
2025

RICARDO AUGUSTO MÜLLER GERMANI

CONVOLUÇÃO INCREMENTAL: Um algoritmo eficiente para a computação de convoluções referentes à funções de equações diferenciais

Projeto de pesquisa submetido ao Curso de Bacharelado em Ciência da Computação do Instituto Federal Sul-Rio-Grandense, Câmpus Passo Fundo, como requisito parcial para a aprovação na disciplina Trabalho de Conclusão de Curso I (TCC I).

Orientador: José Antônio O. de Figueiredo

Coorientador: Denilson José Seidel

PASSO FUNDO

2025

SUMÁRIO

1. INTRODUÇÃO

Diversas áreas do conhecimento se relacionam com o Cálculo Diferencial e Integral através das Equações Diferenciais, que descrevem variáveis com a presença de derivadas e/ou integrais. Para o desenvolvimento de *softwares* que buscam criar uma modelagem de problemas que apresentam as referidas equações, uma série de métodos matemáticos para a resolução de equações diferenciais podem ser utilizados, incluindo a resolução por séries de potência formais - ou seja, representando as funções como séries infinitas - cujo objetivo principal é o cálculo dos respectivos coeficientes.

Por sua vez, a convolução discreta é uma operação matemática realizada entre dois conjuntos de valores *a* e *b*, que resultam em um novo conjunto - descrita pela equação (1) - , e também é aplicável em uma série de tópicos, como, por exemplo, probabilidade, estatística, visão computacional, processamento de sinais e imagens e inclusive equações diferenciais (BEHL et al., 2014).

$$c_n = \sum_{k=-\infty}^{+\infty} a_k \, b_{n-k} \quad (1)$$

A convolução está intimamente ligada, através do teorema da convolução, com as transformadas discretas rápidas, como as FFTs (*Fast Fourier Transform*), possibilitando uma abordagem computacional que prioriza a sua computação mais eficiente, de complexidade temporal log-linear, em detrimento do método ingênuo com complexidade quadrática.

Sua presença está marcada também no Produto de Cauchy (produto entre duas séries infinitas), e consequentemente em equações diferenciais que apresentam o referido produto. Contudo, referente à resolução de equações diferenciais via séries infinitas, e com presença de Produtos de Cauchy, não é possível a utilização direta dos algoritmos convolucionais eficientes - em decorrência da convolução ser calculada de forma incremental, item por item - exigindo o uso de uma outra abordagem, que também deve ser comprovadamente eficiente, no ponto de vista computacional, em relação ao método de resolução eficiente.

Diante deste problema específico, este projeto de pesquisa visa estabelecer uma metodologia para a busca de uma solução que possa resolver o problema destacado e que atenda aos critérios estabelecidos, iniciando a partir de uma pesquisa bibliográfica, com a aplicação do mapeamento sistemático para uma busca inicial de artigos que já apresentem algoritmos eficientes relacionados, seguido pela exploração das propriedades da convolução e de conceitos da programação dinâmica para o desenvolvimento de um algoritmo comprovadamente eficiente para o problema em questão, e a seleção de estudos de casos para a implementação do algoritmo desenvolvido em uma linguagem de programação, possibilitando a validação do algoritmo.

1.1 TEMA

Algoritmos Computacionais e Equações Diferenciais

1.2 PROBLEMA

Como as propriedades da convolução e o conceito da programação dinâmica podem ser aplicados ao desenvolvimento de um algoritmo, computacionalmente eficiente, para resolução de equações diferenciais, que envolvem o produto de funções no formato de séries infinitas.

1.3. OBJETIVOS

1.3.1 Objetivo Geral

Desenvolver um algoritmo eficiente para resolver equações diferenciais com produto de funções em formato de séries infinitas.

1.3.2 Objetivos Específicos

- Mapear algoritmos para resolução de equações diferenciais com produto de funções no formato de séries infinitas;
- Modelar e desenvolver um algoritmo eficiente para resolução de equações diferenciais com produto de funções no formato de séries infinitas, explorando o conceito de programação dinâmica;
- Analisar e provar matematicamente a complexidade computacional do algoritmo proposto;
- Implementar o algoritmo desenvolvido em uma linguagem de programação;
- Validar o algoritmo desenvolvido, testando sua aplicação em estudos de casos selecionados.

1.4. JUSTIFICATIVA

Equações diferenciais consistem em um tópico essencialmente ligado ao Cálculo Diferencial e Integral, por serem responsáveis em estabelecer relações entre variáveis em termos de derivadas e/ou integrais, estando presentes em uma imensa gama de áreas do conhecimento, como Física, Medicina, Química, Biologia, Economia, etc. (SOUSA, 2021). É dividido em dois tipos principais: Equações Diferenciais Ordinárias (EDOs), com uma variável independente; e Equações Diferenciais Parciais (EDPs), com mais de uma variável independente (ROWLETT, [s. d.]). Exemplos para EDOs e EDPs são, respectivamente, as Equações Diferenciais de Bernoulli (1) (DAWKINS, 2003a) e a Equação da Onda (2) (DAWKINS, 2003b):

$$\frac{dy}{dx} + p(x) \cdot y = q(x) \cdot y^{n} \tag{1}$$

$$p(x) \cdot \frac{\partial^2 u}{\partial t^2} = \frac{\partial}{\partial x}\left(T(x,\ t) \cdot \frac{\partial u}{\partial x}\right) + p(x) \cdot Q(x,\ t) \tag{2}$$

Estas equações estão presentes em diversos problemas reais que necessitam ser modelados computacionalmente. Ou seja, em diversas ocasiões, o desenvolvimento de programas de computador que codificam os referidos problemas é necessário, comumente visando calcular os valores das funções de acordo com as equações envolvidas.

Por exemplo, variáveis dependentes do tempo em problemas físicos descritos por equações diferenciais podem ser computadas, iterativamente e de forma aproximada, através de métodos já bastante difundidos na literatura, como, por exemplo, o Método de Euler e o Método de Runge-Kutta (VALLE, 2012). Assim, nota-se que a utilização direta de funções em formato de séries infinitas (séries de potência formais) pode trazer uma outra perspectiva de resolução, através de uma representação matemática bastante fidedigna - dentro do raio de convergência da série em questão. Neste caso, a(s) equação(ões) diferencial(is) podem ser reduzidas apenas para uma expressão/fórmula envolvendo os próprios coeficientes das séries envolvidas, estabelecendo-se uma forma de computá-las de acordo com os coeficientes já computados.

O produto entre funções representadas como séries infinitas resultam em uma nova série, chamada Produto de Cauchy (3), cujo coeficientes correspondem a uma convolução entre os coeficientes das funções originais (SAMBALE, 2024), onde, no contexto das equações diferenciais, cada elemento deve ser computado de forma incremental (em decorrência da citada forma de computar baseada em valores passados), exigindo o uso de um método que use conceitos de programação dinâmica para reduzir as operações aritméticas necessárias. Logo, a busca de uma solução com eficiência computacional melhorada possibilitará a redução de custos computacionais em aplicações que dependem da resolução da(s) função(ões) de uma equação diferencial (ou um sistema delas) como uma série de potência formal, além de trazer uma nova abordagem de resolução deste tipo de problema.

$$\left(\sum_{k=0}^{\infty} a_k\right)\left(\sum_{k=0}^{\infty} b_k\right) = \sum_{k=0}^{\infty}\left(\sum_{j=0}^{k} a_j\, b_{k-j}\right) \tag{3}$$

2. REFERENCIAL TEÓRICO

A pesquisa de trabalhos relacionados buscou seguir o método de mapeamento sistemático descrito por PETERSEN et al. (2015). O mapeamento foi realizado usando os principais portais destinados a artigos acadêmicos: IEEExplore, ACM Digital, SpringerLink, ScienceDirect e Google Acadêmico.

Primeiramente, estabeleceu-se que o objetivo do mapeamento seria a busca de trabalhos que possam apresentar soluções referentes à equações diferenciais que apresentam produtos de funções como séries infinitas, incluindo algoritmos eficientes e/ou exemplos de implementações. Duas questões principais foram definidas para o protocolo: mapear algoritmos eficientes para a resolução de equações diferenciais com Produto de Cauchy; programação dinâmica em algoritmos para resolução de equações diferenciais.

Para as cinco primeiras plataformas citadas anteriormente, fora desenvolvida e utilizada a seguinte *string* de busca: ***(“formal power series” OR “formal power serie” OR “infinite series” OR “infinite serie”) AND “convolution” AND (“fast” OR “efficient”) AND “algorithm” AND (“ODE” OR PDE” OR “differential equations” OR “differential equation)***. Como resultado, foram retornados apenas 3 resultados pelo IEEExplore; 32 resultados pela ACM Digital; 33 artigos pela SpringerLink e 3.680 resultados pelo Google Acadêmico (descontando citações e patentes, e incluindo resultados repetidos nas outras plataformas). Para o ScienceDirect, fora desenvolvida uma string de busca mais enxuta - pela limitação da plataforma - , retornando 118 *research articles*: ***(“formal power series” OR “formal power serie” OR “infinite series” OR “infinite serie”) AND “convolution” AND “algorithm” AND (“ODE” OR “PDE”)***.

Para a seleção dos artigos para leitura, cinco critérios de exclusão foram definidos: 1) Artigo não tem um resumo (*abstract*); 2) Publicação é apenas um resumo (*abstract*); 3) Artigo é uma cópia ou uma versão mais antiga de artigo já considerado; 4) Artigo encontra-se indisponível (sem possível acesso); 5) Artigo não revisado por pares. Após a aplicação dos critérios, 14 artigos foram selecionados a partir do resumo, e 6 artigos foram finalmente destacados após as leituras.

O mapeamento sistemático evidenciou a representação de funções como séries de potência formal (*formal power series*) como uma abordagem já conhecida para a resolução de equações diferenciais, com artigos e livros sobre o tema na literatura - p. ex., SOCHACKI e TONGEN (2022). Além disso, o advento das FFTs (*Fast Fourier Transforms*), que reduziu a complexidade computacional das Transformadas Discretas de Fourier (1) de $O(N^2)$ para

$O(N \, log \, N)$, a partir da década de 1960 possibilitou a rápida computação de convoluções discretas, em decorrência do teorema da convolução (2) (WEISSTEIN, [s. d.]) - ou seja, convoluções, que são calculadas ingenuamente em tempo $O(N^2)$, também poderiam ser resolvidas em $O(N \, log \, N)$:

$$F\{x\}_n = \sum_{k=0}^{N-1} x_k \, e^{-\frac{2\pi}{N}kni} \tag{1}$$

$$F\{x * y\}_n = F\{x\}_n \cdot F\{y\}_n \tag{2}$$

Assim, sabendo que os coeficientes dos Produtos de Cauchy correspondem a uma convolução, surgiram artigos como, por exemplo, o de BRENT e KUNG (1978), que trouxeram métodos rápidos para a manipulação de séries de potência formal no contexto de equações diferenciais.

A convolução também pode apresentar-se como uma operação contínua (em forma de integral) em diferentes tipos de equações diferenciais, e dado a enorme complexidade para a busca de soluções analíticas na maioria dos casos, os pesquisadores buscam discretizar as operações, ou seja, equações que apresentam convoluções contínuas são aproximadas através de convoluções discretas, ao assumir as funções como séries de potência formal - um exemplo é o método da “Convolução Singular Discreta”, descrito por WEI(2002).

Dentro deste contexto, propriedades da convolução discreta podem ser aproveitadas em conjunto com o teorema da convolução, possibilitando um uso sistemático de transformadas discretas para a criação de um algoritmo com melhor eficiência computacional: um dos artigos mais notáveis é o de HAIRER, LUBICH e SCHLICHTE (1985), que apresenta uma solução eficiente para a resolução de equações de Volterra não lineares, aproveitando a quadratura da convolução para a computação de uma convolução discreta em uma complexidade temporal $O(N \, log^2 \, N)$ para os N primeiros termos; três anos depois, em 1988, os mesmos pesquisadores publicaram uma nova solução (de mesma complexidade temporal) para equações de Volterra “fracamente singulares”, incluindo um código da solução na linguagem Fortran.

Os trabalhos de Harier et. al. serviram como referência para diversos trabalhos posteriores: trabalhos como os de DIETHELM (2006) e GARRAPPA (2018) usam as soluções já desenvolvidas como base para a resolução de Equações Diferenciais Fracionárias (FDEs, na sigla em inglês) no formato de integral de Riemann-Liouville. Há também artigos que abordam implementações em uma linguagem de programação, como o artigo de KHALIGHI et. al. (2024), que apresenta um algoritmo Julia de código aberto destinado à resolução de FDEs, referenciando os artigos de Garrappa.

3. METODOLOGIA

De acordo com WAINER (2007, p. 1), “[a] pesquisa em Ciência da Computação [...] envolve na maioria dos casos a construção de um programa, de um modelo, de um algoritmo ou de um sistema novo”, o que corresponde exatamente ao objetivo deste projeto de pesquisa, que é o desenvolvimento de um algoritmo eficiente para solucionar o problema definido. Observa-se também que o tipo de pesquisa a ser realizado é qualitativo.

A pesquisa será dividida em cinco etapas principais, detalhadas no restante deste capítulo. São elas: 1) Desenvolvimento de um algoritmo eficiente, baseado nas propriedades da convolução e em princípios da programação dinâmica; 2) Análise do custo computacional do algoritmo desenvolvido; 3) Implementação do algoritmo desenvolvido em uma determinada linguagem de programação; 4) Validação do algoritmo; e 5) Análise dos resultados obtidos.

3.1 DESENVOLVIMENTO DE UM ALGORITMO EFICIENTE

Na primeira etapa da pesquisa, será realizado o desenvolvimento de um algoritmo que possibilite a computação dos coeficientes das séries infinitas correspondentes às funções presentes em uma equação diferencial ou em um sistema de equações diferenciais que apresente, pelo menos, um Produto de Cauchy. A condição principal é que o algoritmo a ser desenvolvido apresente uma eficiência computacional melhorada em relação ao algoritmo ingênuo, aproveitando características da operação da convolução (presente nos Produtos de Cauchy) para que a computação possa ser baseada em transformadas discretas, de acordo com o teorema da convolução. A programação dinâmica também será fundamental no processo de desenvolvimento, a partir da divisão do problema maior - o cálculo da “convolução incremental” - em sub-problemas que podem ser calculados e armazenados para etapas posteriores (COELHO, 2021).

Considerando que os coeficientes a serem computados estejam restritos exclusivamente ao Conjunto dos Números Reais, o foco no desenvolvimento do algoritmo será na aplicação de transformadas discretas voltadas para valores reais, como, por exemplo, a Transformada Discreta de Hartley (FHT, na sigla em inglês) (1), descrita por BRACEWELL (1983):

$$X_n = \sum_{k=0}^{N-1} x_k \, cas\left(\frac{2\pi kn}{N}\right) = \sum_{k=0}^{N-1} x_k \left(cos\left(\frac{2\pi kn}{N}\right) + \, sin\left(\frac{2\pi kn}{N}\right)\right) \quad (1)$$

3.2 ANÁLISE DO CUSTO COMPUTACIONAL DO ALGORITMO DESENVOLVIDO

Após a construção do algoritmo, será necessário desenvolver uma demonstração matemática da complexidade computacional temporal tanto do algoritmo ingênuo quanto do algoritmo proposto, para que se possa comparar as complexidades obtidas, de modo a validar a eficiência do novo algoritmo perante o algoritmo ingênuo. Além disso, também será desenvolvido um pseudocódigo do algoritmo proposto, que pode ser utilizado por outros estudantes e/ou pesquisadores para a sua implementação em qualquer linguagem de programação.

3.3 IMPLEMENTAÇÃO DO ALGORITMO DESENVOLVIDO

Uma vez que o desenvolvimento do novo algoritmo, junto com a comprovação de sua eficácia, foi finalizado, alguns estudos de casos serão selecionados para serem codificados através de uma linguagem de programação de escolha do autor - para este projeto, a linguagem escolhida foi C++. Os estudos de caso serão focados em exemplos práticos que apresentam Equações Diferenciais Ordinárias (EDOs), ou seja, baseada em apenas uma variável independente (que, para os estudos de caso, corresponderá ao tempo) Os estudos de caso estão detalhados na subseção 5.3.1.

Para cada estudo de caso, serão implementados, pelo menos, dois códigos: um código correspondente ao algoritmo ingênuo para a computação dos coeficientes das séries infinitas das funções; e outro correspondente ao algoritmo proposto. Outro(s) código(s) pode(m) ser implementado(s) com o uso de outras técnicas de resolução de EDOs para o cálculo das variáveis dependentes, como, por exemplo, o Método de Euler ou o Método de Runge-Kutta. Considerando que as equações diferenciais referentes aos estudos de caso são baseadas em uma variável temporal, as implementações devem ser capazes de criar uma simulação das variáveis dependentes com o decorrer do tempo.

As implementações obtidas buscam atender dois objetivos: confirmar a eficiência computacional do novo algoritmo comparando o tempo de processamento das suas respectivas implementações com o tempo executado pelos códigos do algoritmo ingênuo, de acordo com a quantidade de coeficientes a serem computados; além disso, é decerto que uma quantidade maior de coeficientes equivale a representar as funções por um polinômio de maior grau, mais próximo do valor definitivo (da série infinita) e garantindo uma maior precisão das simulações, portanto uma convergência dos resultados das simulações, de acordo com a quantidade de coeficientes, também deverá ser observada.

3.3.1 Estudos de caso previstos para a validação

Atendendo às condições previamente postas, os principais estudos de casos que foram selecionados para esta etapa são os seguintes:

- **Gravitação Newtoniana**: A clássica teoria gravitacional descrita pelo físico inglês Isaac Newton em *Principia* (1687) descreve que a força gravitacional entre dois corpos é diretamente proporcional ao produto de suas massas e inversamente proporcional ao quadrado da distância entre seus centros de massa. Considerando que o espaço real apresenta três dimensões, o movimento realizado por dois corpos de acordo com a gravidade newtoniana é descrito através de um sistema de três EDOs, que, assumindo que as massas sejam constantes, relacionam as suas posições com suas acelerações (derivada de segunda ordem). Em um contexto unidimensional simples, a função posição da fórmula da gravitação newtoniana (NEWTON, 1999) transforma-se na solução de uma EDO (2):

$$F = -\frac{G \cdot M_A \cdot M_B}{S^2} \rightarrow M_A \cdot \frac{d^2S}{dt^2} = -\frac{G \cdot M_A \cdot M_B}{S^2} \qquad (2)$$

- **Equação do Arrasto**: Atribuída ao físico inglês John William Strutt (Lord Rayleigh), a referida equação indica que quando um corpo atravessa um determinado fluído, uma força de resistência exercida pelo fluído é aplicada sobre o corpo, cujo módulo é diretamente proporcional ao quadrado da velocidade do corpo, e também à densidade do fluído e área de contato. Assumindo que as últimas duas variáveis (mais a massa do corpo) seja constante, as derivadas de primeira (velocidade) e segunda (aceleração) ordem da posição do corpo podem ser diretamente relacionadas, através de uma equação diferencial ordinária. BENSON (1999) descreve a equação do arrasto (3) da seguinte forma, denotando C_d como coeficiente de arrasto, r densidade, v velocidade e A área de contato, que pode ser transformada na seguinte equação diferencial:

$$F = \frac{C_d \cdot r \cdot v^2 \cdot A}{2} \rightarrow M \cdot \frac{d^2S}{dt^2} = \frac{C_d \cdot r \cdot A}{2} \left(\frac{dS}{dt}\right)^2 \qquad (3)$$

3.4 VALIDAÇÃO DO ALGORITMO

Antes de testar os códigos implementados, exemplos para os estudos de casos devem ser previamente estabelecidos, ou seja, os valores iniciais das variáveis envolvidas (como, por exemplo, massa e velocidade), assim como a quantidade de coeficientes a serem calculados para os algoritmos baseados em séries infinitas, devem ser informados antes que a simulação seja realizada. Uma vez definidos os exemplos, os códigos deverão ser executados, visando simular cada cenário por um determinado intervalo para a variável de tempo.

Para a próxima etapa da pesquisa, uma série de informações deverão ser coletadas durante a execução dos testes e armazenados em arquivos de log: para cada teste, será salvo no seu respectivo arquivo uma lista contendo os valores das variáveis dependentes de acordo com cada instante de tempo observado, e também, para os algoritmos baseados em séries infinitas, o tempo de processamento de cada simulação junto com a quantidade de coeficientes calculados, com um “teto” definido para cada problema abordado.

3.5 ANÁLISE DOS RESULTADOS OBTIDOS

Após a execução dos testes, o aluno será responsável, na última etapa da pesquisa, pela análise das informações armazenadas nos arquivos de log gerados. Para cada exemplo de teste definido, o aluno deverá reunir os dados gerados por todos os códigos implementados, e fazer uso de uma planilha eletrônica (como, por exemplo, Google Planilhas) para tabulação e visualização dos dados obtidos: o primeiro gráfico deve apresentar os valores das variáveis computadas (eixo y) em função da variável de tempo (eixo x), dentro do intervalo de tempo estabelecido para a simulação em si; e o segundo deve apresentar o tempo de processamento dos algoritmos (eixo *y*) baseados em séries infinitas em função da quantidade de coeficientes computados (eixo *x*).

A primeira análise deverá observar a convergência dos valores das variáveis computadas pelos algoritmos baseados em séries infinitas (tanto o algoritmo ingênuo, quanto o algoritmo proposto) em decorrência da quantidade de coeficientes computados, além de também comparar com os valores computados por outro algoritmo de resolução de equações diferenciais. Já na segunda análise, deve-se avaliar a eficiência do algoritmo proposto em relação ao algoritmo ingênuo, além da relação dos gráficos gerados com suas respectivas complexidades computacionais.

Com a finalização do projeto, deverá ser disponibilizado através de um *link* público os códigos implementados para os estudos de caso selecionados.

3.6 REQUISITOS DO PROJETO

A pesquisa exigirá apenas um Ambiente de Desenvolvimento Integrado (IDE), compatível com a linguagem de programação C++ (como, por exemplo, CodeBlocks ou Visual Studio Code), para a codificação dos códigos necessários, além de uma plataforma *online* que possibilita a geração de gráficos (como, por exemplo, Google Planilhas).

4. CRONOGRAMA

Atividades	**Ago**	**Ago**	**Set**	**Set**	**Out**	**Out**	**Nov**	**Nov**
Desenvolvimento do algoritmo (pseudocódigo e prova matemática)	X	X						
Implementação estudo de caso 1			X	X				
Implementação estudo de caso 2					X			
Validação dos resultados (testes)						X	X	
Documentação dos resultados	X	X	X	X	X	X	X	
Redação do artigo	X	X	X	X	X	X	X	X
Entrega do trabalho								X

5. REFERÊNCIAS

BEHL, Akshay; BHATIA, Akash; PURI, Avril. Convolution and Applications of Convolution. 2014 IJIRT, Volume 1, Issue 6, p. 2122 - 2126. Disponível em: <https://ijirt.org/publishedpaper/IJIRT101029_PAPER.pdf>. Acesso em: 26 jun 2025.

BENSON, Tom. The Drag Equation. NASA, 1999. Disponível em: <https://web.archive.org/web/20000816193526/https://www.grc.nasa.gov/WWW/K-12/airplane/drageq.html>. Acesso em: 24 jun 2025.

BRACEWELL, Ronald N. Discrete Hartley transform. Journal of the Optical Society of America, Volume 7, 1983. Disponível em: <https://scispace.com/papers/discrete-hartley-transform-4hx6b5c9ov?citations_page=38>. Acesso em: 26 jun 2025.

BRENT, Richard P.; KUNG, H. T. Fast Algorithms for Manipulating Formal Power Series. Association for Computing Machinery, Volume 25, Issue 4, p. 581 - 595, 1978. Disponível em: <https://dl.acm.org/doi/pdf/10.1145/322092.322099>. Acesso em: 23 jun 2025.

COELHO, Thiago Henrique Neves. Programação dinâmica na prática: Do básico ao intermediário. 2021. (Trabalho de Conclusão de Curso - Monografia), Curso de Bacharelado em Ciência da Computação, Universidade Federal do Rio de Janeiro, Rio de Janeiro - Brasil, 2021. Disponível em: <https://pantheon.ufrj.br/bitstream/11422/14161/1/THNCoelho.pdf>. Acesso em: 26 jun 2025.

DAWKINS, Paul. Section 2.4: Bernoulli Differential Equations. Paul's Online Notes, 2003a. Disponível em: <https://tutorial.math.lamar.edu/Classes/DE/Bernoulli.aspx>. Acesso em: 26 jun 2025.

DAWKINS, Paul. Section 9.2: The Wave Equation. Paul's Online Notes, 2003b. Disponível em: <https://tutorial.math.lamar.edu/Classes/DE/TheWaveEquation.aspx>. Acesso em: 26 jun 2025.

DIETHELM, Kai; FORD, Judith M.; FORD, Neville J.; WEILBEER, Marc. Pitfalls in fast numerical solvers for fractional differential equations. Journal of Computation and Applied Mathematics, Volume 186, Issue 2, p. 482 - 503, 2006. Disponível em: <https://www.sciencedirect.com/science/article/pii/S037704270 5001287#bib26>. Acesso em: 17 jun 2025.

GARRAPPA, Roberto. Numerical Solution of Fractional Differential Equations: A Survey and a Software Tutorial. Fractional Calculus: Theory and Applications. *Mathematics 2018*, Volume 6, Issue 2, 2018. Disponível em: <https://www.mdpi.com/2227-7390/6/2/16>. Acesso em: 29 mai 2025.

HAIRER, Ernst; LUBICH, Ch.; SCHLICHTE, M. Fast Numerical Solution of Nonlinear Volterra Convolution Equations. SIAM Journal on Scientific and Statistical Computing, Volume 6, Issue 3, p. 532 - 541, 1985. Disponível em: <https://www.researchgate.net/publication/246169389_Fast_Numerical_Solution_of_Nonlinear_Volterra_Convolution_Equations>. Acesso em: 29 mai 2025.

HAIRER, Ernst; LUBICH, Ch.; SCHLICHTE, M. Fast numerical solution of weakly singular Volterra integral equations. Journal of Computational and Applied Mathematics, Volume 23, Issue 1, p. 87 - 98, 1988. Disponível em: <https://www.sciencedirect.com/science/article/pii/0377042788903329>. Acesso em: 29 mai 2025.

KHALIGHI, Moein; BENEDETTI, Giulio; LAHTI, Leo. Algorithm 1047: FdeSolver, a Julia Package for Solving Fractional Differential Equations. ACM Transactions on Mathematical Software, Volume 50, Issue 3. Article No.: 22, p. 1 - 23. Disponível em: <https://dl.acm.org/doi/10.1145/3680280>. Acesso em: 17 jun 2025.

NEWTON, Isaac. The Principia: Preceded by a Guide to Newton's Principia by I. Bernard Cohen: Mathematical Principles of Natural Philosophy. Estados Unidos: University of California Press, 1999. Disponível em: <https://archive.org/download/NewtonPrincipia_201701/The%20Principia%20Mathematical%20Principles%20of%20Natural%20Philosophy%20Isaac%20Newton.pdf>. Acesso em: 24 jun 2025.

PETERSEN, Kai; VAKKALANKA, Sairam; KUZNIARZ, Ludwik. Guidelines for conducting systematic mapping studies in software engineering: An update. Information and Software Technology, Volume 64, p. 1 - 18, 2015. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0950584915000646>. Acesso em: 15 mai 2025.

ROWLETT, Julie. Partial and ordinary differential equations and systems. Disponível em: <https://www.math.chalmers.se/Math/Grundutb/CTH/mve030/1617/mve290-PDEs.pdf>. Acesso em: 26 jun 2025.

SAMBALE, B. An Invitation to Formal Power Series. *Jahresber. Dtsch. Math*, Volume 125, p. 3 - 69, 2023. Disponível em: <https://doi.org/10.1365/s13291-022-00256-6>. Acesso em: 08 mai 2025.

SOCHACKI, James. TONGEN, Anthony. Applying Power Series to Differential Equations. Textbook, 2022. Disponível em: <https://link.springer.com/book/10.1007/978-3-031-24587-9>. Acesso em: 22 mai 2025.

SOUSA, Damião Franceilton Marques de. Equações diferenciais ordinárias aplicadas na física clássica. 2021. (Trabalho de Conclusão de Curso – Monografia), Curso de Licenciatura em Física, Centro de Educação e Saúde, Universidade Federal de Campina Grande, Cuité - Paraíba - Brasil, 2021. Disponível em: <http://dspace.sti.ufcg.edu.br:8080/xmlui/bitstream/handle/riufcg/20691/DAMI%C3%83O%20FRANCEILTON%20MARQUES%20DE%20SOUSA%20-%20TCC%20LICENCIATURA%20EM%20F%C3%8DSICA%20CES%202021.pdf?sequence=1&isAllowed=y>. Acesso em: 08 mai 2025.

VALLE, Karine Nayara F. Métodos Numéricos de Euler e Runge-Kutta. 2012. (Trabalho de Conclusão de Curso - Monografia), Especialização em Educação Matemática, Universidade Federal de Minas Gerais, Belo Horizonte - Minas Gerais - Brasil, 2012. Disponível em: <https://repositorio.ufmg.br/bitstream/1843/48530/1/Metodos%20de%20Euler%20e%20Runge%20Kutta.pdf>. Acesso em: 08 mai 2025.

WAINER, Jacques. Métodos de pesquisa quantitativa e qualitativa para a Ciência da Computação. Sociedade Brasileira de Computação/Editora PUC Rio, Volume 1, 2007.

Disponível em: <https://edisciplinas.usp.br/pluginfile.php/4922788/mod_resource/content/1/WainerPesquisaCC.pdf>. Acesso em: 24 jun 2025.

WEI, G. W.; ZHAO, Y. B.; XIANG, Y. Discrete singular convolution and its application to the analysis of plates with internal supports. Part 1: Theory and algorithm. International Journal for Numerical Methods in Engineering, Volume 55, Issue 8, p. 913 - 946, 2002. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.526?casa_token=wvPP0Bwj1lcAAAAA:wCGbbhUsNsX11dfFXLv1irhdCepN5onRq64yFlmPyphbWW5r6M2X2btBYrIyu7LiVGrD3gV-YMdU7X0a>. Acesso em: 17 jun 2025.

WEISSTEIN, Eric W. Convolution Theorem. MathWord - A Wolfram Web Resource. Disponível em: <https://mathworld.wolfram.com/ConvolutionTheorem.html>. Acesso em: 26 jun 2025.