https://doi.org/10.23925/2237-9657.2025.v14i1p153-157

# Derivadas e aproximações: o caso da função $f(x) = e^x$ e a aproximação afim $\ell(x) = x + 1$

HUMBERTO JOSÉ BORTOLOSSI[1]

https://orcid.org/0000-0003-1212-6252

LUCIANA PRADO MOUTA PENA[2]

https://orcid.org/0000-0003-1259-6473

**RESUMO**

*Este texto explora, de uma perspectiva numérica/gráfica no GeoGebra, o uso da função afim* $\ell(x) = x+1$ *para aproximar a função incluindo valores de* $x$ *não tão próximos de* $p = 0$.

**Palavras-chave:** *aproximação de funções; Cálculo diferencial; GeoGebra.*

## Derivatives and Approximations: the case of the function $f(x) = e^x$ and the affine approximation $\ell(x) = x + 1$.

**ABSTRACT**

*This text explores, from a numerical/graphical perspective in GeoGebra, the use of the affine function* $\ell(x) = x+1$ *to approximate the function* $f(x) = e^x$ *including values of* $x$ *that are not so close to* $p = 0$.

**Keywords:** *Function approximation; differential calculus; GeoGebra.*

## Introdução

Nos cursos de Cálculo I aprende-se que derivadas permitem construir funções mais simples de funções deriváveis (eventualmente mais complicadas). Vejamos um exemplo clássico: considere a função exponencial $f(x) = e^x$, com $e = 2{,}718281828459045...$ a constante de Euler (um número irracional, base dos logaritmos naturais)[3] .

Avaliar $f(x) = e^x$ para diferentes valores numéricos de $x$ pode constituir um desafio. Entram em jogo, questões de precisão, rapidez e das operações que o *hardware* de um computador pode fazer (essencialmente versões estritas das quatro operações aritméticas). Nos cursos de Cálculo mostramos que a linearização[4]

[1] Doutor em Matemática, Pontifícia Universidade Católica do Rio de Janeiro, PUC-Rio. Professor, Universidade Federal Fluminense (UFF), campus Niterói- humbertobortolossi@id.uff.br

[2] Doutora em Modelagem Computacional e Matemática Aplicada, Universidade do Estado do Rio de Janeiro (UERJ). Professora, Universidade Federal Fluminense (UFF), campus Niterói, Rio de Janeiro, Brasil - lucianapena@id.uff.br

[3] A universalidade da exponencial tanto na matemática pura quanto aplicada fez com que o renomado matemático Walter Rudin a chamasse da "função mais importante da Matemática".

[4] Um certo abuso frequente, já que $\ell$ não é exatamente linear. Mas muito natural: o gráfico da exponencial perto de $x$ é trocado por uma

$$\ell(x) = \mathrm{f}(p) + f'(p)(x - p)$$

é a melhor aproximação afim da função f próximo ao ponto p. Certamente, a função $\ell$ é muito mais fácil de avaliar para diferentes valores de $x$. Aqui, melhor está no sentido de minimizar uma expressão para o erro de aproximação, o que pode ser estabelecido pelo Teorema de Taylor. Vamos aos detalhes: para uma função $f$ diferenciável em um intervalo aberto $I$, com $p \in I$, pelo Teorema de Taylor, para qualquer $x \in I$, temos:

$$f(x) = \underbrace{f(p) + f'(p)(x - p)}_{\ell(x)} + R(x)$$

onde $f(p)$ e $f'(p)$ são os valores de $f$ e sua derivada $f'$ em $p$, e o resto $R(x) = f(x) - \ell(x)$ mede o erro de aproximação. Mais ainda, existe uma função $h$ tal que

$$R(x) = f(x) - \ell(x) = h(x)(x - p), \text{com } lim_{x \to p} h(x) = 0.$$

Considere, agora, qualquer outra função afim $g(x) = m(x - p) + f(p)$ cujo gráfico (uma reta!) passe pelo ponto $(p, f(p))$ do gráfico de $f$, aqui $m$ é um número real que determina a inclinação da reta, ou seja, seu coeficiente angular. O erro de aproximação da função $f(x)$ por essa outra função afim $g$ em um ponto $x$ é dado por:

$$E(x) = \left| f(x) - \left[ \underbrace{m(x - p) + f(p)}_{g(x)} \right] \right| \quad (1)$$

Substituindo em (1), obtemos:

$$E(x) = \left| \underbrace{f(p) + f'(p)(x - p)}_{\ell(x)} + R(x) - [m(x - p) + f(p)] \right|$$

simplificando a expressão, obtemos:

$$E(x) = |[f'(p) - m](x - p) + R(x)|.$$

Vemos então que o erro de aproximação será minimizado quando a inclinação $m$ da reta for igual à inclinação da tangente, ou seja, quando $m = f'(p)$. Neste caso, o erro de aproximação se resume ao termo de resto $R(x)$, que tende a zero quando $x$ tende a $p$. Em conclusão, o Teorema de Taylor demonstra que a reta tangente ao gráfico da função $f$ no ponto $(p, f(p))$ é a melhor aproximação afim da função nesse ponto, no sentido de minimizar o erro de aproximação.

Em nosso exemplo, $f(x) = e^x$. Vamos escolher $p = 0$, um ponto onde é fácil calcular $f(x) = e^x$ e sua derivada: $f'(x) = f(x) = e^x$. Temos, assim:

$$p = 0,\ f'(0) = f(0) = e^0 = 1 \text{ e } \ell(x) = f(0) + f'(0)(x - 0) = 1 + x.$$

Como pode ser visto se $x$ é próximo de $p = 0$ então $\ell(x)$ é próximo de $f(x)$.

reta, o gráfico de $\ell$.

Vejamos um exemplo numérico:

$$e^{0.01} = f(0.01) \approx \ell(0.01) = 1 + 0.01 = 1.01$$

Uma calculadora científica comum fornece: $e^{0.01} = 1.01005016\ldots$ A aproximação afim dá três casas decimais corretas, sem contar que o cálculo da função $y = \ell(x)$ usa apenas uma única operação aritmética.

## 2. Aproximações para outros valores de x

Vamos agora ver o que acontece quando $x$ não está próximo de $p = 0$, por exemplo, $x = 1$. Nesse caso, é suficiente considerar que:

$$e^1 = (e^{0.5})^2 = ((e^{0.25})^2)^2 = (((e^{0.125})^2)^2)^2 = ((((e^{0.0625})^2)^2)^2)^2 = \cdots.$$

Procedendo com esta ideia chegamos a, por exemplo,

$$e^1 = (((((((((((e^{0.00048828125})^2)^2)^2)^2)^2)^2)^2)^2)^2)^2)^2$$

Agora, 0,00048828125 é um número próximo de $0$ e podemos, assim, usar a aproximação afim $\ell(x) = 1 + x$:

$$e^{0.00048828125} \approx 1 + 0.00048828125 = 1.00048828125$$

Substituindo esta aproximação na expressão anterior, obtemos que

$$e^1 \approx (((((((((((e^{1.00048828125})^2)^2)^2)^2)^2)^2)^2)^2)^2)^2)^2 \approx 1.00048828125^{2048}$$

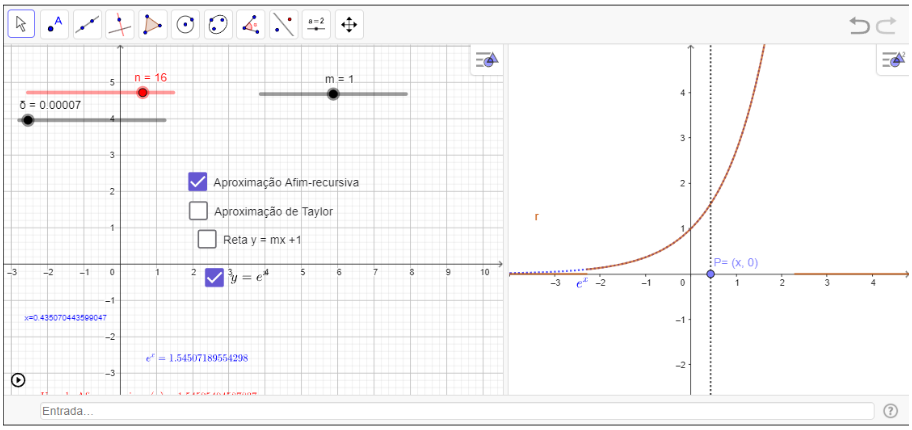
o que nos dá $e^1 \approx 2.7176184823368797609$. Uma calculadora científica fornece $e^1 \approx 2.7176184823368796$. Note que estamos usando um procedimento recursivo:

$$r(x) = \text{se } (x < 0.001 \text{ retorne } 1 + x\text{, caso contrário retorne } (r(\mathrm{x}/2))^2).$$

## 2. *Applet* GeoGebra

A técnica descrita na seção anterior foi implementada no *applet* GeoGebra disponível em https://www.geogebra.org/m/g3xus7jv (Figura 1). Nesse *applet*, o controle deslizante $n$ determina a profundidade de recursão da função $r$, o controle deslizante $\delta$ determina o intervalo $(-\delta, \delta)$ no qual a aproximação afim $\ell(x) = 1 + x$ é usada. Além da aproximação afim-recursiva, o *applet* exibe o gráfico de $f(x) = e^x$ como desenhado nativamente pelo *software* GeoGebra.O polinômio de Taylor de ordem $k$ também é exibido para fim de comparação. Por fim, ainda é possível comparar numericamente os valores de todas essas funções para diferentes valores de $x$. O valor de $x$ pode ser alterado movendo-se o ponto de nome $P$.

**Figura 1:** *Applet* GeoGebra com aproximação afim-recursiva da função $f(x) = e^x$



**Fonte:** Os autores [5], 2023.

A função recursiva $r$ foi definida usando-se o comando Iteração do GeoGebra: Iteração $\left(\text{Se}\left(\text{abs}(x)<\delta, 1+m\,x, \left(g\left(\frac{x}{2}\right)\right)^2\right), g, \{f\}, n\right)$, onde $g(x)=x^2$.

## Considerações Finais

O IEEE (*Institute of Electrical and Electronics Engineers*) é uma das principais organizações internacionais responsáveis pela definição de padrões técnicos amplamente adotados na indústria da computação e engenharia elétrica. No entanto, o IEEE não especifica diretamente uma única forma de implementar, em *hardware*, o cálculo de funções elementares como a exponencial $f(x)=e^x$, deixando essa escolha a critério de cada fabricante de CPU. Como resultado, diversos algoritmos têm sido propostos para esse fim:

1. **Aproximações polinomiais ou racionais:** Muitos algoritmos utilizam aproximações polinomiais ou racionais de funções elementares. Isso é especialmente útil para funções que não podem ser calculadas diretamente de forma exata, como seno, cosseno e logaritmos.
2. **Tabelas pré-calculadas:** Algumas CPUs utilizam tabelas de valores pré-calculados combinados com métodos de correção para refinar o valor da função. Isso é feito para acelerar o cálculo de funções elementares complexas, reduzindo o número de operações aritméticas necessárias.
3. **Algoritmos CORDIC** (*Coordinate Rotation Digital Computer*): Amplamente utilizado para calcular funções trigonométricas e hiperbólicas com eficiência, o CORDIC também pode ser empregado no cálculo de potências e logaritmos. Seu funcionamento baseia-se em iterações simples que utilizam apenas operações de adição e deslocamentos de bits, evitando a necessidade de multiplicações ou divisões. Essas operações tornam o algoritmo especialmente adequado para implementações em *hardware*, como FPGAs[6] e DSPs[7], onde a economia de recursos computacionais é crucial.

[5] Disponível em: <https://www.geogebra.org/m/g3xus7jv >. Acesso em 08 ago. 2025.
[6] FPGA (*Field-Programmable Gate Array*) é um dispositivo semicondutor programável composto por blocos lógicos configuráveis e interconexões reprogramáveis, permitindo a implementação de circuitos digitais personalizados diretamente no hardware.
[7] *DSP* (*Digital Signal Processor*) é um processador especializado, projetado para executar operações matemáticas de forma muito rápida e

4. **Iterações de Newton-Raphson:** Esse método iterativo é usado para cálculos como divisão e raiz quadrada, fornecendo alta precisão em poucas iterações.
5. **Métodos de fatoração e decomposição:** Para funções como $f(x) = e^x$, a decomposição em fatores menores facilita a aproximação em valores maiores de $x$[8], o que melhora a precisão do cálculo com menos operações.

Para um panorama geral destas técnicas, indicamos a referência (MULLER, 2006).

No estudo de Detrey, Dinechin e Pujol (2007), os autores observam que:

> O estudo de circuitos de hardware específicos para a avaliação de funções elementares em ponto flutuante já foi uma área de pesquisa ativa, até que se percebeu que essas funções não eram frequentes o suficiente para justificar a dedicação de silício a elas. A pesquisa, então, voltou-se para funções implementadas em software. No entanto, essa situação pode estar prestes a mudar novamente com o advento de coprocessadores reconfiguráveis baseados em FPGAs (Field-Programmable Gate Arrays). Esses coprocessadores agora têm uma capacidade que permite acomodar cálculos em ponto flutuante de dupla precisão. Operadores de *hardware* para funções elementares, direcionados a essas plataformas, têm o potencial de superar significativamente as funções em *software* e não desperdiçarão permanentemente os recursos de silício. (DETREY; DINECHIN & PUJOL, 2007, p. 161)

## Agradecimentos

Os autores agradecem aos professores Carlos Tomei e Marco Moriconi pela leitura minuciosa, sugestões e correções do texto original.

## Referências

DETREY, J.; DINECHIN, F.; PUJOL, X. *Return of the hardware floating-point elementary function*, 1**8th Symposium on Computer Arithmetic**, Montpellier, France, pp.161-168, 2007. Disponível em: <https://ens-lyon.hal.science/ensl-00117386v1/document>. Acesso em: 08 ago. 2025.

MULLER., J. M. **Elementary Functions: Algorithms and Implementation**, Second Edition, New York: Birkhäuser, 2006.

## ENVIADO: 08/02/2025

## ACEITO: 12/08/2025

eficiente, especialmente em aplicações que envolvem processamento de sinais digitais, como áudio, vídeo e telecomunicações.

[8] Observe que esta foi a técnica que empregamos em nosso exemplo.