FEV 3-5 2026
I WORKSHOP MINEIRO DE MATEMÁTICA APLICADA
UFMG - BELO HORIZONTE - MG

# Explorando alguns efeitos dos erros de Ponto Flutuante

**Ribas, E. R. L. D.[1,*], Pazini, D. S.[2], D'Afonseca, L. A.[3], Rocha, L. M.[4]**

**Departamento de Matemática, Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)**

***Contato: enzorochaleitedinizribas@gmail.com**

## Introdução

A *aritmética de ponto flutuante* é o sistema adotado por computadores para que lidem com números reais utilizando uma notação compacta e eficaz. Essa técnica é utilizada para representar e manipular números reais de forma prática e eficiente. Ela permite representar números de grandezas diversas, que não podem ser armazenados com precisão, utilizando apenas números inteiros.

A aritmética de ponto flutuante é amplamente utilizada em diversas áreas, como computação científica, gráficos de computador, simulações numéricas e processamento de sinais. No entanto, é importante compreender suas limitações e os possíveis erros que podem ocorrer durante as operações aritméticas, a fim de garantir resultados precisos e confiáveis em cálculos numéricos. Os resultados a seguir são derivados de estudos realizados em um projeto de iniciação científica.

## Ponto Flutuante

Um sistema de ponto flutuante $F$ pode ser definido como

$$F(\beta, t, L, U)$$

cuja representação normalizada de um número real N nesse sistema é dada por

$$N = \pm(d_1.d_2...d_t)_\beta \times \beta^e \tag{1}$$

em que

- $N$ é o número real;
- $\beta$ é a base que a máquina opera;
- $t$ é o número de dígitos na mantissa, tal que $0 \leq d_j \leq \beta - 1$, j = 1, ...,t, $d_1 \neq 0$;
- $L$ é o menor expoente inteiro;
- $U$ é o maior expoente inteiro;
- $e$ é o expoente inteiro no intervalo [$L$,$U$].

## Metodologia

Para investigar os efeitos dos erros de ponto flutuante, foram realizados experimentos computacionais utilizando diferentes configurações de precisão numérica. Através da linguagem de programação Python junto a bibliotecas NumPy e Scipy para melhor confiabilidade das operações e outras bibliotecas gráficas como Matplotlib, seaborn, plotly, foram implementados algoritmos que reproduzem instabilidades em operações aritméticas.

## Fenômenos Observados

Um erro comum neste sistema é o da perda de significância, ou cancelamento catastrófico, que ocorre quando a subtração de dois números resulta em um valor com menos dígitos significativos do que os números originais. Para ilustrar isso consideramos duas situações.

Na Figura 1, a função $f(x) = x^{10} + 1 - x^{10}$, embora para $x \in \mathbb{R}$ o resultado seja igual a 1, ao efetuarmos os cálculos usando ponto flutuante, observamos um comportamento inesperado em que o valor correto é exibido apenas até um certo valor de $x$. Após esse valor observamos um intervalo em que ocorre uma oscilação no resultado da função e, em seguida, observamos que a função assume o valor zero.

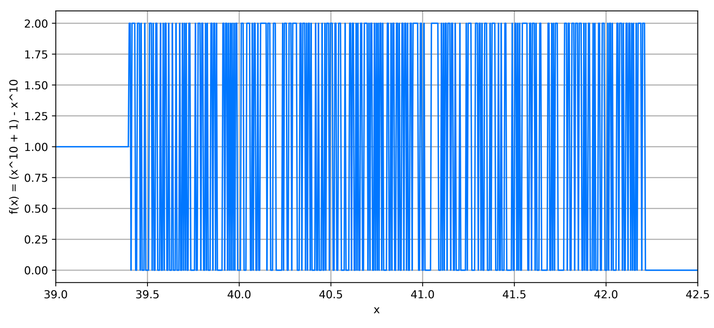


**Figura 1:** Perda de significância com de 32 bits.

Já na Figura 2, consideramos a função $f(x) = \frac{(1+x)-1}{x}$, que matematicamente é igual a 1 para todo $x \neq 0$. Contudo, ao calcular essa função utilizando ponto flutuante, para valores de $x$ muito pequenos, também observamos um comportamento oscilatório.

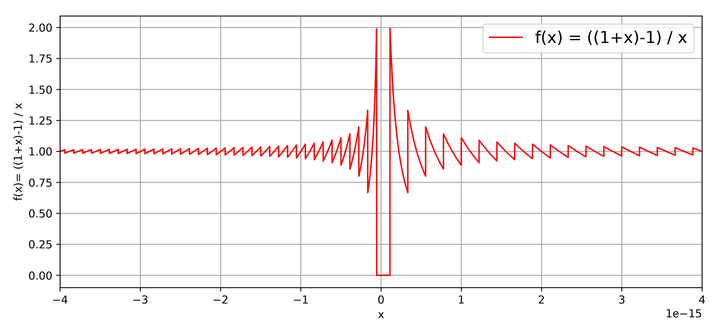


**Figura 2:** Erro de cancelamento catastrófico com precisão de 64 bits.

Um outro erro é o não cancelamento adequado de termos em expressões matematicamente equivalentes. A Figura 3 apresenta a comparação entre os resultados obtidos ao calcular $p(x) = (x-1)^6$ e sua forma expandida $q(x) = x^6 - 6x^5 + 15x^4 - 20x^3 + 15x^2 - 6x + 1$. Nesta figura, o gráfico de $p$ exibe os resultados obtidos pelo cálculo da expressão fatorada utilizando ponto flutuante obtendo a curva esperada. Entretanto, o gráfico da expressão expandida $q$, calculado no mesmo sistema de ponto flutuante, produz resultados caóticos. Note que, apesar de possivelmente surpreendente, esse fenômeno não invalida a utilidade do ponto flutuante, pois o erro observado é da ordem de $10^{-14}$.

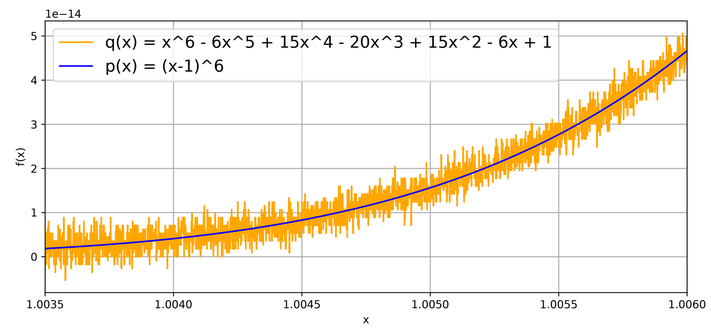


**Figura 3:** Gráfico das funções p e q com precisão de 64 bits.

Também observamos erros em expressões racionais onde o numerador e o denominador ficam pequenos simultaneamente. Utilizando Diferenças finitas para aproximar a derivada de uma função, ao diminuir o valor de $h$, espera-se que a aproximação melhore. Mas devido aos erros de arredondamento em ponto flutuante, para h muito pequeno, o erro na aproximação começa a se comportar de forma caótica. A Figura 4 ilustra esse fenômeno para a função $f(x) = \frac{x^3}{3} - 3x + 3$ cuja derivada é dada por $f'(x) = x^2 - 3$. O valor aproximado da derivada é calculado pelo quociente

$$D_f(x,h) = \frac{f(x+h) - f(x-h)}{2h}$$

A derivada numérica via diferença central é então comparada com o valor exato da derivada em $x = 1$, e o erro é definido como

$$Erro(h) = |f'(1) - D_f(1,h)|$$

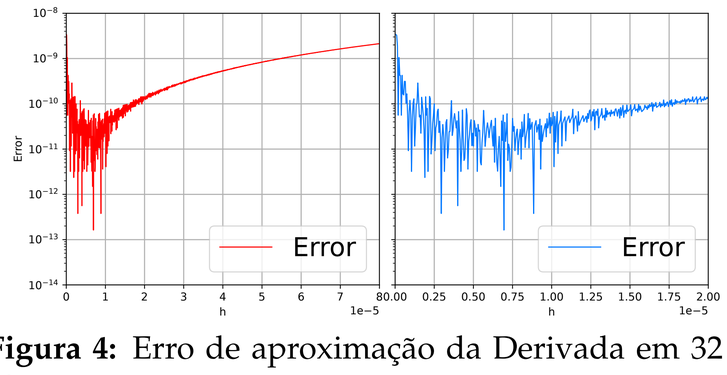


**Figura 4:** Erro de aproximação da Derivada em 32 bits.

## Referências

[1] Chapra, S. C., Canale, R. P. Numerical Methods for Engineers. McGraw-Hill International Editions, 1985.

[2] IEE Standard for Floating-Point Arithmetic. IEEE Std 754-2019, 2019.

[3] Faires J. D., Burden R. L. Numerical Analysis. 7th Edition. Brooks/Cole, 2001.

[4] Ruggiero, M. A. G., Lopes, V. L. R. Cálculo Numérico: Aspectos Teóricos e Computacionais. 2th Edition. Pearson. 2010.