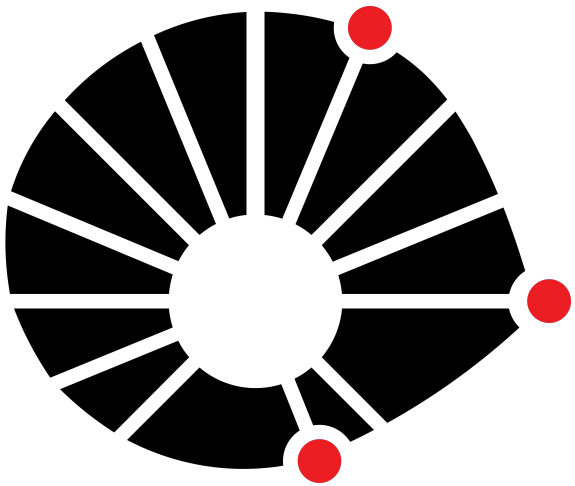
MS211 - Cálculo Numérico

Aula 2 – Erros e Aritmética de Números em Ponto Flutuante.



UNICAMP

Marcos Eduardo Valle
Matemática Aplicada
IMECC - Unicamp

Representação em Ponto Flutuante

Definição 1 (Sistema de Ponto Flutuante)

Um número real $x \neq 0$ é um ponto flutuante (normalizado) se pode ser escrito como

$$x = \pm 0.d_1 d_2 d_3 \ldots d_t \times \beta^e,$$

em que

- β é a base;
- t é o número de dígitos na mantissa, com $d_1 \neq 0$ e $0 \leqslant d_j \leqslant \beta - 1$, para todo $j = 1, \ldots, t$.
- e é o exponente, com $-m \leqslant e \leqslant M$.

Denotamos por $F(\beta, t, m, M)$ o conjunto de todos os pontos flutuantes para β, t, m e M fixos e adicionando algumas exceções como o zero, `Inf` e `NaN`.

Exemplo 2

Considere o sistema $F(10, 3, 2, 2)$. Represente nesse sistema, se possível, os números:

$$x_1 = 0.35, \quad x_2 = -5.17, \quad x_3 = 0.0123, \tag{1}$$

$$x_4 = 5390, \quad x_5 = 0.0003. \tag{2}$$

Exemplo 2

Considere o sistema $F(10, 3, 2, 2)$. Represente nesse sistema, se possível, os números:

$$x_1 = 0.35, \quad x_2 = -5.17, \quad x_3 = 0.0123, \tag{1}$$
$$x_4 = 5390, \quad x_5 = 0.0003. \tag{2}$$

Resposta:

$$x_1 = 0.350 \times 10^0, \quad x_2 = -0.517 \times 10^1, \quad x_3 = 0.123 \times 10^{-1}.$$

O número $5390 = 0.539 \times 10^4$ não pode ser representado porque seu expoente é maior que 2. Tem-se *overflow*.
O número $0.0003 = 0.300 \times 10^{-3}$ não pode ser representado porque seu exponente é menor que -2. Tem-se um *underflow*.

A maioria dos computadores trabalha com a base $\beta = 2$. Esse não é um problema, pois qualquer base terá suas limitações.

A maioria dos computadores trabalha com a base $\beta = 2$. Esse não é um problema, pois qualquer base terá suas limitações.

Veja, por exemplo no livro texto “M. Ruggiero e V. Lopes. Cálculo Numérico - Aspectos Teóricos e Computacionais, 2ª edição, Editora Pearson, 1997”, como é feita a mudança de base!

A maioria dos computadores trabalha com a base $\beta = 2$. Esse não é um problema, pois qualquer base terá suas limitações.

Veja, por exemplo no livro texto “M. Ruggiero e V. Lopes. Cálculo Numérico - Aspectos Teóricos e Computacionais, 2ª edição, Editora Pearson, 1997”, como é feita a mudança de base!

Muitos *softwares* científicos usam o padrão IEEE **precisão dupla** com 64 bits: 1 para o sinal, 11 para o expoente, 53 para a mantissa, resultando no sistema $F(2, 53, 1.022, 1.023)$.

A maioria dos computadores trabalha com a base $\beta = 2$. Esse não é um problema, pois qualquer base terá suas limitações.

Veja, por exemplo no livro texto “M. Ruggiero e V. Lopes. Cálculo Numérico - Aspectos Teóricos e Computacionais, 2ª edição, Editora Pearson, 1997”, como é feita a mudança de base!

Muitos *softwares* científicos usam o padrão IEEE **precisão dupla** com 64 bits: 1 para o sinal, 11 para o expoente, 53 para a mantissa, resultando no sistema $F(2, 53, 1.022, 1.023)$.

O padrão IEEE precisão dupla é capaz de representar números positivos entre 1.79×10^{308} e 2.23×10^{-308}, aproximadamente.

A maioria dos computadores trabalha com a base $\beta = 2$. Esse não é um problema, pois qualquer base terá suas limitações.

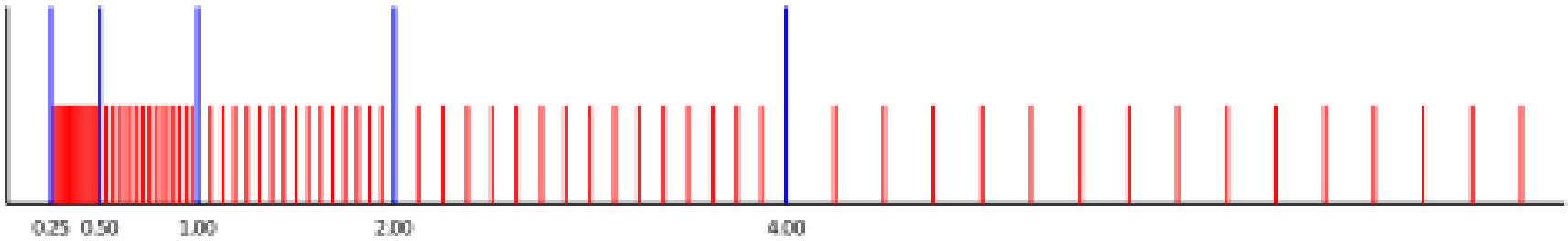
Veja, por exemplo no livro texto "M. Ruggiero e V. Lopes. Cálculo Numérico - Aspectos Teóricos e Computacionais, 2ª edição, Editora Pearson, 1997", como é feita a mudança de base!

Muitos *softwares* científicos usam o padrão IEEE **precisão dupla** com 64 bits: 1 para o sinal, 11 para o expoente, 53 para a mantissa, resultando no sistema $F(2, 53, 1.022, 1.023)$.

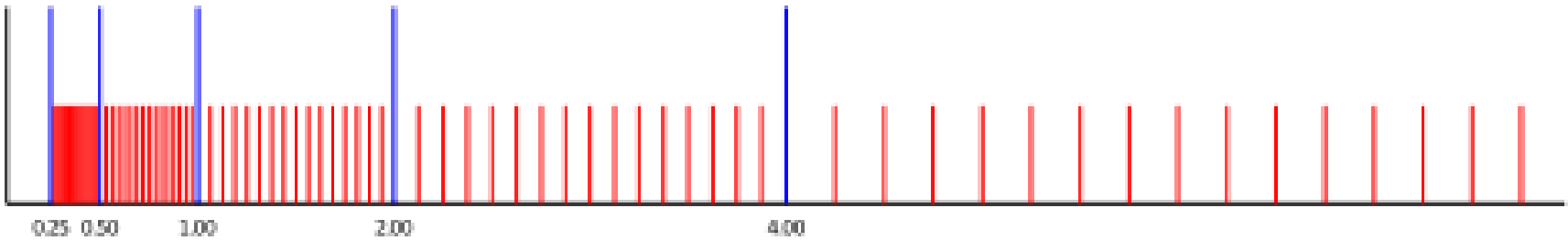
O padrão IEEE precisão dupla é capaz de representar números positivos entre 1.79×10^{308} e 2.23×10^{-308}, aproximadamente.

O padrão IEEE possui uma representação especial para o zero, `Inf` (obtido após a divisão por zero), e `NaN` (Not a Number, e.g. $0/0$).

A figura abaixo ilustra a parte positiva do conjunto de pontos flutantes $F(2,2,2,2)$:

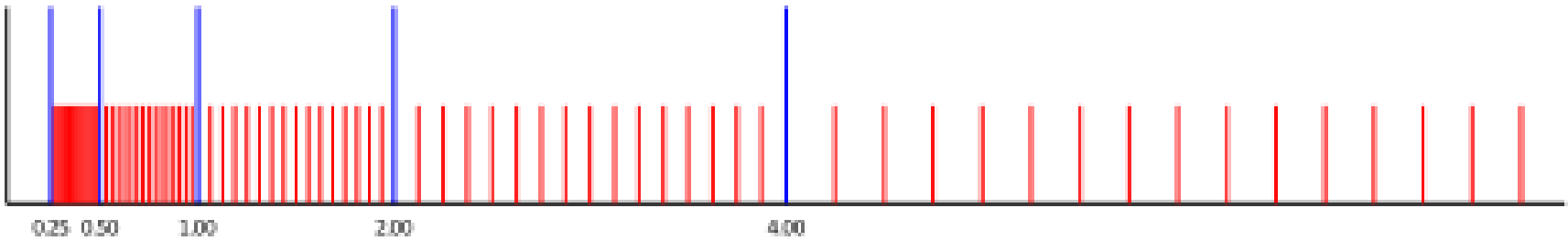


A figura abaixo ilustra a parte positiva do conjunto de pontos flutantes $F(2, 2, 2, 2)$:



É importante observar que, independente da base β, a quantidade de pontos flutuantes entre β^e e β^{e+1} é a mesma, para qualquer exponente e.

A figura abaixo ilustra a parte positiva do conjunto de pontos flutantes $F(2, 2, 2, 2)$:



É importante observar que, independente da base β, a quantidade de pontos flutuantes entre β^e e β^{e+1} é a mesma, para qualquer exponente e.

Consequentemente, as lacunas entre os pontos flutuantes aumentam conforme aumenta o exponente.

Arredondamento em Ponto Flutuante

O arredondamento em ponto flutuante é usado para representar um número real x, dentro dos limites de representação do sistema, que não pertence ao conjunto $F(\beta, t, m, M)$.

Arredondamento em Ponto Flutuante

O arredondamento em ponto flutuante é usado para representar um número real x, dentro dos limites de representação do sistema, que não pertence ao conjunto $F(\beta, t, m, M)$.

Especificamente, arredondar um número x em ponto flutuante consiste em encontrar $\bar{x} \in F(\beta, t, m, M)$ tal que $|x - \bar{x}|$ seja o menor possível.

Arredondamento em Ponto Flutuante

O arredondamento em ponto flutuante é usado para representar um número real x, dentro dos limites de representação do sistema, que não pertence ao conjunto $F(\beta, t, m, M)$.

Especificamente, arredondar um número x em ponto flutuante consiste em encontrar $\bar{x} \in F(\beta, t, m, M)$ tal que $|x - \bar{x}|$ seja o menor possível.

Denotaremos por `fl` a função que associa um número real x ao seu arredondamento em ponto flutuante, ou seja, $\bar{x} = \texttt{fl}(x)$.

Arredondamento em Ponto Flutuante

O arredondamento em ponto flutuante é usado para representar um número real x, dentro dos limites de representação do sistema, que não pertence ao conjunto $F(\beta, t, m, M)$.

Especificamente, arredondar um número x em ponto flutuante consiste em encontrar $\bar{x} \in F(\beta, t, m, M)$ tal que $|x - \bar{x}|$ seja o menor possível.

Denotaremos por `fl` a função que associa um número real x ao seu arredondamento em ponto flutuante, ou seja, $\bar{x} = \texttt{fl}(x)$.

O valor $|x - \bar{x}|$ é chamado **erro absoluto** de arredondamento.

De um modo similar, o valor $\dfrac{|x - \bar{x}|}{|x|}$ é chamado **erro relativo** de arredondamento.

Exemplo 3

Represente no sistema $F(10, 3, 5, 5)$ os números

$$x_1 = 1234.56, \quad x_2 = -0.00054962, \quad x_3 = 0.9995,$$
$$x_4 = 123456.7, \quad x_5 = 0.0000001.$$

Exemplo 3

Represente no sistema $F(10, 3, 5, 5)$ os números

$$x_1 = 1234.56, \quad x_2 = -0.00054962, \quad x_3 = 0.9995,$$
$$x_4 = 123456.7, \quad x_5 = 0.0000001.$$

Resposta:

$$\texttt{fl}(x_1) = 0.123 \times 10^4, \quad \texttt{fl}(x_2) = -0.550 \times 10^{-3},$$
$$\texttt{fl}(x_3) = 0.100 \times 10^1.$$

Exemplo 3

Represente no sistema $F(10, 3, 5, 5)$ os números

$$x_1 = 1234.56, \quad x_2 = -0.00054962, \quad x_3 = 0.9995,$$
$$x_4 = 123456.7, \quad x_5 = 0.0000001.$$

Resposta:

$$\texttt{fl}(x_1) = 0.123 \times 10^4, \quad \texttt{fl}(x_2) = -0.550 \times 10^{-3},$$
$$\texttt{fl}(x_3) = 0.100 \times 10^1.$$

Para x_4 e x_5 tem-se *overflow* e *underflow*, respectivamente.

Exemplo 3

Represente no sistema $F(10, 3, 5, 5)$ os números

$$x_1 = 1234.56, \quad x_2 = -0.00054962, \quad x_3 = 0.9995,$$
$$x_4 = 123456.7, \quad x_5 = 0.0000001.$$

Resposta:

$$\texttt{fl}(x_1) = 0.123 \times 10^4, \quad \texttt{fl}(x_2) = -0.550 \times 10^{-3},$$
$$\texttt{fl}(x_3) = 0.100 \times 10^1.$$

Para x_4 e x_5 tem-se *overflow* e *underflow*, respectivamente.

Para arredondar um número na base $\beta = 10$, devemos apenas observar o primeiro dígito a ser descartado. Se ele for menor que 5, deixamos os dígitos inalterados; Se ele é maior ou igual a 5, devemos somar 1 ao último dígito remanescente.

Épsilon da Máquina

Definição 4 (Épsilon da Máquina)

O épsilon da máquina, denotado por ε_{mach}, é a distância entre 1 e o menor ponto flutuante estritamente maior que 1.

Pode-se mostrar que o épsilon de uma máquina de um sistema $F(\beta, t, m, M)$ é

$$\varepsilon_{mach} = 0.10\ldots01 \times \beta^1 - 0.10\ldots00 \times \beta^1 = \beta^{1-t}.$$

Épsilon da Máquina

Definição 4 (Épsilon da Máquina)

O épsilon da máquina, denotado por ε_{mach}, é a distância entre 1 e o menor ponto flutuante estritamente maior que 1.

Pode-se mostrar que o épsilon de uma máquina de um sistema $F(\beta, t, m, M)$ é

$$\varepsilon_{mach} = 0.10\ldots01 \times \beta^1 - 0.10\ldots00 \times \beta^1 = \beta^{1-t}.$$

No padrão IEEE precisão dupla, a precisão é

$$\varepsilon_{mach} = 2^{-52} \approx 2.2 \times 10^{-16}.$$

O ε_{mach} fornece um limitante superior para o erro relativo do arredondamento em ponto flutuante.

O ε_{mach} fornece um limitante superior para o erro relativo do arredondamento em ponto flutuante.

Especificamente, para qualquer x dentro dos limites de representação do sistema, existe $\bar{x} \in F(\beta, t, m, M)$ tal que

$$\left| \frac{x - \bar{x}}{x} \right| \leqslant \varepsilon_{mach}.$$

O ε_{mach} fornece um limitante superior para o erro relativo do arredondamento em ponto flutuante.

Especificamente, para qualquer x dentro dos limites de representação do sistema, existe $\bar{x} \in F(\beta, t, m, M)$ tal que

$$\left| \frac{x - \bar{x}}{x} \right| \leqslant \varepsilon_{mach}.$$

Esta última inequação resulta na seguinte afirmação:

Proposição:

Para qualquer número real x dentro dos limites de representação do sistema, existe ε com $|\varepsilon| \leqslant \varepsilon_{mach}$ tal que

$$\texttt{fl}(x) = x(1 + \varepsilon).$$

Aritmética de Ponto Flutuante

Além de representar números no computador, precisamos também efetuar operações com eles.

Aritmética de Ponto Flutuante

Além de representar números no computador, precisamos também efetuar operações com eles.

As operações aritméticas básicas $+, -, \times$ e $\div$ com números reais, quando realizadas no computador com sistema $F(\beta, t, m, M)$, serão denotadas por $\oplus$, $\ominus$, $\otimes$ e $\oslash$.

Aritmética de Ponto Flutuante

Além de representar números no computador, precisamos também efetuar operações com eles.

As operações aritméticas básicas $+, -, \times$ e $\div$ com números reais, quando realizadas no computador com sistema $F(\beta, t, m, M)$, serão denotadas por $\oplus$, $\ominus$, $\otimes$ e $\oslash$.

As operações aritméticas de ponto flutuante são definidas de modo a satisfazer o axioma:

Axioma das Operações de Ponto Flutuante:

Sejam $*$ uma operação aritmética básica e $\circledast$ seu análogo em ponto flutuante. Para todo $x, y \in F(\beta, t, m, M)$, deve-se ter

$$x \circledast y = \texttt{fl}(x * y).$$

Em vista do axioma das operações de ponto flutuante, tem-se:

Proposição:

Para quaisquer $x, y \in F(\beta, t, m, M)$, existe ε com $|\varepsilon| \leqslant \varepsilon_{mach}$ tal que

$$x \circledast y = (x * y)(1 + \varepsilon),$$

em que $*$ denota uma operação aritmética básica e $\circledast$ seu análogo em ponto flutuante.

Em vista do axioma das operações de ponto flutuante, tem-se:

Proposição:

Para quaisquer $x, y \in F(\beta, t, m, M)$, existe ε com $|\varepsilon| \leqslant \varepsilon_{mach}$ tal que

$$x \circledast y = (x * y)(1 + \varepsilon),$$

em que $*$ denota uma operação aritmética básica e $\circledast$ seu análogo em ponto flutuante.

Essa proposição estabelece uma relação entre a operação aritmética com números reais e seu análogo em ponto flutuante.

Em vista do axioma das operações de ponto flutuante, tem-se:

Proposição:

Para quaisquer $x, y \in F(\beta, t, m, M)$, existe ε com $|\varepsilon| \leqslant \varepsilon_{mach}$ tal que

$$x \circledast y = (x * y)(1 + \varepsilon),$$

em que $*$ denota uma operação aritmética básica e $\circledast$ seu análogo em ponto flutuante.

Essa proposição estabelece uma relação entre a operação aritmética com números reais e seu análogo em ponto flutuante. Essa relação possui um papel importante na análise de erros de algoritmos!

Em vista do axioma das operações de ponto flutuante, tem-se:

Proposição:

Para quaisquer $x, y \in F(\beta, t, m, M)$, existe ε com $|\varepsilon| \leqslant \varepsilon_{mach}$ tal que

$$x \circledast y = (x * y)(1 + \varepsilon),$$

em que $*$ denota uma operação aritmética básica e $\circledast$ seu análogo em ponto flutuante.

Essa proposição estabelece uma relação entre a operação aritmética com números reais e seu análogo em ponto flutuante. Essa relação possui um papel importante na análise de erros de algoritmos!

Por ora, gostaríamos de destacar que as operações de ponto flutuante não gozam de todas as propriedades das operações dos números reais!

Exemplo 5 (Associatividade e Distributividade)

Considere o sistema $F(10, 3, 5, 5)$. Sejam $x = \texttt{fl}(11.4)$, $y = \texttt{fl}(3.18)$ e $z = \texttt{fl}(5.06)$. Efetue as operações:

(a) $(x \oplus y) \oplus z$ e $x \oplus (y \oplus z)$.

(b) $\dfrac{y \otimes x}{z}$ e $\dfrac{y}{z} \otimes x$.

(c) $y \otimes (z \oplus x)$ e $(y \otimes z) \oplus (y \otimes x)$.

Exemplo 5 (Associatividade e Distributividade)

Considere o sistema $F(10, 3, 5, 5)$. Sejam $x = \texttt{fl}(11.4)$, $y = \texttt{fl}(3.18)$ e $z = \texttt{fl}(5.06)$. Efetue as operações:

(a) $(x \oplus y) \oplus z$ e $x \oplus (y \oplus z)$.

(b) $\dfrac{y \otimes x}{z}$ e $\dfrac{y}{z} \otimes x$.

(c) $y \otimes (z \oplus x)$ e $(y \otimes z) \oplus (y \otimes x)$.

Resposta:

(a) $(x \oplus y) \oplus z = 0.197 \times 10^2$e $x \oplus (y \oplus z) = 0.196 \times 10^2$.

(b) $\dfrac{x \otimes y}{z} = 0.717 \times 10^1$ e $x \otimes \left(\dfrac{y}{z}\right) = 0.716 \times 10^1$.

(c) $y \otimes (z \oplus x) = 0.525 \times 10^2$ e $(y \otimes z) \oplus (y \otimes x) = 0.524 \times 10^2$.

Ao contrário das operações com números reais, as operações de ponto flutuante não são nem associativas e nem distributivas!

Exemplo 6 (Identidade Não-Nula)

Introduzimos os seguintes comandos:

```
a,b = 1,1
while a+b > a:
   b = b/2
```

Estaríamos num *loop* infinito se fizéssemos as mesmas operações com números reais!

Exemplo 6 (Identidade Não-Nula)

Introduzimos os seguintes comandos:

```
a,b = 1,1
while a+b > a:
   b = b/2
```

Estaríamos num *loop* infinito se fizéssemos as mesmas operações com números reais!

Num computador, porém, encontramos

$$b = 1.1102 \times 10^{-16} = \frac{\varepsilon_{mach}}{2}.$$

Exemplo 6 (Identidade Não-Nula)

Introduzimos os seguintes comandos:

```
a,b = 1,1
while a+b > a:
   b = b/2
```

Estaríamos num *loop* infinito se fizéssemos as mesmas operações com números reais!

Num computador, porém, encontramos

$$b = 1.1102 \times 10^{-16} = \frac{\varepsilon_{mach}}{2}.$$

Note que

```
a+b = a,
```

ou seja, existe na aritmética de ponto flutuante um número $b \neq 0$ tal que $a + b = a$.

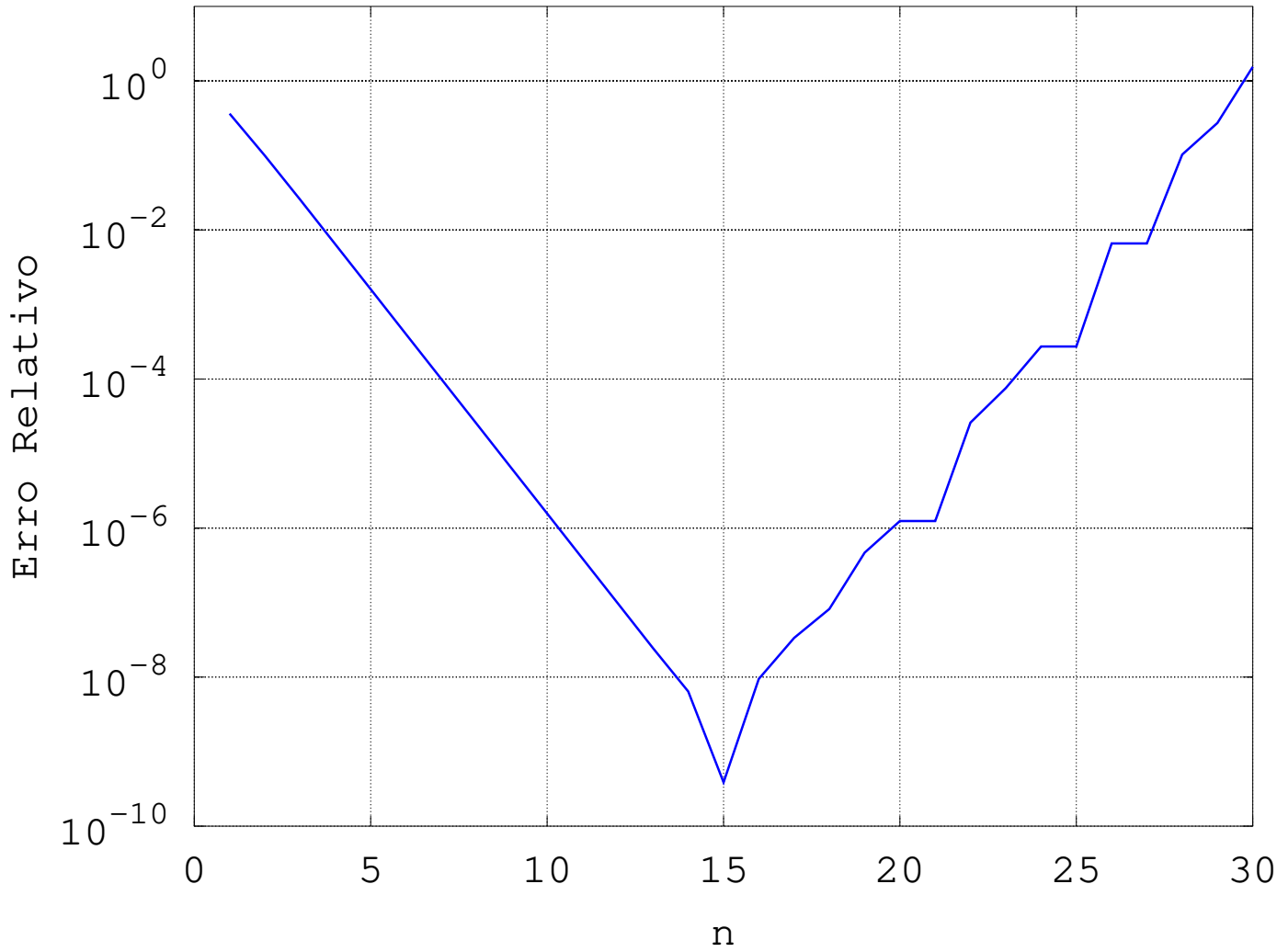
Exemplo 7 (Estimando o valor de π)

Considere a seguinte sequencia de comandos que, teoricamente, forneceria uma estimativa para o número π:

```
z = [2]
for n in range(2,30):
    aux = 1-math.sqrt(1-(z[-1]**2)*4**(1-n))
    z.append((2**(n-0.5))*math.sqrt(aux))
```

O seguinte gráfico apresenta o erro relativo das estimativas para π obtido pelo comando:

```
[abs(t-math.pi)/math.pi for t in z]
```



Note que o erro decresce nas primeiras 16 iterações mas depois cresce devido aos erros de arredondamento.

Exemplo 8 (Cancelamento Numérico)

Sabemos que a identidade

$$\frac{(1+x)-1}{x} = 1,$$

para todo $x \neq 0$.
No `python`, porém, encontramos o seguinte:

```
x = 1.e-15
((1+x)-1)/x
1.1102230246251565
```

Note que temos um erro relativo superior a 11% devido ao cancelamento de algarismos significativos.

Considerações Finais

Na aula de hoje, apresentamos a representação dos números nos computadores usuais e destacamos erros que podem surgir na aritmética de pontos flutuantes.

Considerações Finais

Na aula de hoje, apresentamos a representação dos números nos computadores usuais e destacamos erros que podem surgir na aritmética de pontos flutuantes.

Os erros de arredondamento, quando repetidos em algoritmos longos e complexos, podem ter efeitos cadastróficos. Exemplos incluem:

- Fracasso do míssil *Patriot* durante a Guerra do Golfo em 1991, devido a um erro de arredondamento no cálculo de sua trajetória.
- Explosão do míssil Arianne em Junho de 1996 devido à *overflow* no computador de bordo.

Muito grato pela atenção!