

ALC: Erros Numéricos

Prof. Alexandre Salles da Cunha e Profa. Ana Paula Couto

2023-1



UNIVERSIDADE FEDERAL
DE MINAS GERAIS



- $f_1(x) = \frac{e^x - e^{-2x}}{x}, \quad \lim_{x \rightarrow 0} f_1(x) = 3$

- $f_2(x) = \frac{x - \sin(x)}{x^3}, \quad \lim_{x \rightarrow 0} f_2(x) = \frac{1}{6}$

Implementação das funções na forma como as lemos...

```
n = 18
x = np.zeros(n, dtype = 'float64')
f1 = np.zeros(n, dtype = 'float64')
f2 = np.zeros(n, dtype = 'float64')
for i in range(n):
    x[i] = np.power(0.1, i+1)
    f1[i] = (math.exp(x[i]) - math.exp(-2.*x[i]))/x[i]
    f2[i] = (x[i] - math.sin(x[i]))/(np.power(x[i], 3.0))
```

x =	f_1 =	f_2 =
0.1	2.8644016	0.1665834
0.0100000	2.9851494	0.1666658
0.0010000	2.9985015	0.1666667
0.0001000	2.99985	0.1666667
0.0000100	2.999985	0.1666673
0.0000010	2.9999985	0.1666537
0.0000001	2.9999998	0.1720536
1.000D-08	3.	0.
1.000D-09	3.	0.
1.000D-10	3.0000002	0.
1.000D-11	3.0000002	0.
1.000D-12	3.0000447	0.
1.000D-13	2.9987124	0.
1.000D-14	2.9976022	0.
1.000D-15	3.1086245	0.
1.000D-16	2.220446	0.
1.000D-17	0.	0.
1.000D-18	0.	0.

Expansão de $\sin(x)$ em série de potências

$$\begin{aligned}\sin(x) &= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \\ &= \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)!}\end{aligned}$$

Função re-escrita

$$\begin{aligned}f_2(x) &= \frac{1}{x^3} \left(x - \left(x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \right) \right) \\ &= \frac{1}{6} - \sum_{n=2}^{\infty} (-1)^n \frac{x^{2(n-1)}}{(2n+1)!}\end{aligned}$$

```
def AvaliaF2(x,kmax,tol):  
    v = 1.0/6.0  
    k = 2  
    convergiu = 0  
    while (k <= kmax) & (convergiu == 0):  
        f = math.factorial(2*k+1)  
        termo = np.power(-1,k)*np.power(x,2*(k-1))/f  
        v = v - termo  
        if abs(termo) <= tol:  
            convergiu = 1  
    print('k = ',k,'v = ', "{:.10f}".format(v),\  
          '|termo| = ', "{:.10E}".format(abs(termo)))  
    k = k + 1
```

AvaliaF2(.5,8,1.0e-10)

k = 2 v = 0.1645833333 |termo| = 2.0833333333E-03

k = 3 v = 0.1645957341 |termo| = 1.2400793651E-05

k = 4 v = 0.1645956911 |termo| = 4.3058311287E-08

k = 5 v = 0.1645956912 |termo| = 9.7859798381E-11

AvaliaF2(1.0E-16,8,1.0e-10)

k = 2 v = 0.1666666667 |termo| = 8.3333333333E-35

Mudança de base - $18,1875_{10} = ?_2$

- 1 Transformação da parte inteira: $18_{10} = ?_2$ $18_{10} = 10010_2$
- 2 Transformação da parte fracionária: $,1875_{10} = ?_2$
 $,1875_{10} = ,0011_2$
- 3 Logo $18,1875_{10} = 10010,0011_2$

$$0,6000_{10} = ?_2$$

Mudança de base - $18,1875_{10} = ?_2$

- 1 Transformação da parte inteira: $18_{10} = ?_2$ $18_{10} = 10010_2$
- 2 Transformação da parte fracionária: $,1875_{10} = ?_2$
 $,1875_{10} = ,0011_2$
- 3 Logo $18,1875_{10} = 10010,0011_2$

$$0,6000_{10} = ?_2$$

$$,6000_{10} = ,10011001..._2$$

Representação normalizada

$$\pm, d_1 d_2 \dots d_p \times B^e$$

onde:

- p é o número de dígitos da mantissa
- B é base, alfabeto = $\{0, 1, \dots, B - 1\}$,
- $d_i : i = 1, \dots, p$ são os dígitos da mantissa
- $d_i \in \{0, 1, \dots, B - 1\}$
- $d_1 \neq 0$
- expoente e é um inteiro na faixa $e \in [-l, u]$

Indicaremos o resultado de uma expressão y por $fl(y)$ ou *float* de y .

Computador hipotético

Base decimal, $B = 10$, $l = u = 5$, $p = 2$ dígitos.

① $4,32 + 0,064$

② $1234 \times 0,016$

③ 875×3172

④ $0,0064 / 7312$

⑤ $372 - 371$

⑥ $691 + 2,71$

Formalmente, a *precisão da máquina* é o menor número ϵ no formato 2^{1-p} para algum p inteiro positivo, tal que $fl(1 + \epsilon) > fl(1)$.

```
def PrecisaodaMaquina():  
    a = 1.0  
    u = 1.0  
    b = a + u  
    k = 0  
    while (b != a):  
        u = u / 2.0  
        b = a + u  
        k = k + 1  
    r = a + u  
    p = 2*u  
    print('r = ',r,'k = ',k,'Precisao = ', "{:.16E}".format(p))  
    return r,k,p
```

```
r,k,p = PrecisaodaMaquina()  
r = 1.0 k = 53 Precisao = 2.2204460492503131E-16
```

```
print(np.finfo('float64').eps)
print(np.finfo('float').eps)
print(np.finfo('float32').eps)
```

2.220446049250313e-16

2.220446049250313e-16

1.1920929e-07

```
v = (1.0 + 0.5*np.finfo('float64').eps) - 1.0
print(v)
0.0
```