Trabalho Prático 1

Grupo 37

October 29, 2021

Introdução

O objetivo do trabalho é calcular valores aproximados de somas infinitas, com erro absoluto inferior a um valor ϵ dado, por majoração do erro de truncatura. Pretende-se analisar a eficácia dos métodos utilizados, bem como a exatidão dos resultados obtidos. A linguagem de programação utilizada foi Python.

Epsilon de máquina

Para calcular o epsilon de máquina utilizou-se o programa

```
eps=1
while eps +1 != 1:
    eps /= 2

print(eps*2)
```

obtendo-se o valor $eps = 2.220446049250313e - 16$.

Exercício 2

$$S = \frac{9}{2\sqrt{3}} \sum_{k=0}^{\infty} \frac{k!^2}{(2k+1)!} \tag{1}$$

S é uma série de termos positivos. Seja $a_n = \frac{9}{2\sqrt{3}} \frac{n!^2}{(2n+1)!}$. Verifica-se que

$$\lim_{n\to\infty} \frac{a_{n+1}}{a_n} = \frac{\frac{9}{2\sqrt{3}} \frac{(n+1)!^2}{(2n+3)!}}{\frac{9}{2\sqrt{3}} \frac{n!^2}{(2n+1)!}} = \frac{(n+1)^2}{(2n+3)(2n+2)} = \frac{1}{4} < 1 \tag{2}$$

logo, a série é convergente.

De facto, tem-se que $\frac{a_{n+1}}{a_n} < \frac{1}{4}$, para todo o n. Assim, aplica-se o critério de D'Alembert.

Seja $S_n = \frac{9}{2\sqrt{3}} \sum_{k=0}^{n-1} \frac{k!^2}{(2k+1)!}$ a soma parcial dos n primeiros termos de S e seja $R_n = S - S_n$. Tem-se:

$$|R_n| \leq a_n \frac{1}{1 - \frac{1}{4}} = \frac{4a_n}{3} = \frac{6}{\sqrt{3}} \frac{n!^2}{(2n+1)!} \tag{3}$$

Logo, basta encontrar n tal que $\frac{6}{\sqrt{3}} \frac{n!^2}{(2n+1)!} < \epsilon$ e somar os primeiros n termos da série. O programa utilizado foi:

```
from math import *

def a(k):
    return 9/(2*sqrt(3))*factorial(k)**2/factorial(2*k+1)
```

```
def soma(i):
    erro = 10**(-i)
    n=0
    while (6/sqrt(3))*factorial(n)**2/factorial(2*n+1) >= erro:
        n+=1

    s=0
    for k in range(n):
        s += a(k)

    return(erro, n,s)

for i in range(8,16):
    print(soma(i))
```

Obtêm-se os seguintes valores:

ϵ	n	S
10^{-8}	14	3.1415926506432688
10^{-9}	15	3.1415926528764797
10^{-10}	17	3.141592653547753
10^{-11}	19	3.1415926535873
10^{-12}	20	3.14159265358918478
10^{-13}	22	3.1415926535897567
10^{-14}	23	3.1415926535897842
10^{-15}	25	3.1415926535897927

CONCLUSÕES

Exercício 3

$$S = 4\sum_{k=0}^{\infty}\frac{(-1)^k}{(2k+1)} \tag{4}$$

S é uma série alternada. Seja $a_n = 4\frac{(-1)^n}{2n+1}$. Como $|a_n|$ é decrescente e $\lim_{n\to\infty}|a_n| = 0$, S é convergente.

Seja $S_n = 4\sum_{k=0}^{n-1}\frac{(-1)^k}{(2k+1)}$ a soma parcial dos n primeiros termos de S e seja $R_n = S - S_n$. Tem-se:

$$R_n \leq |a_n| \tag{5}$$

Logo, basta encontrar n tal que $|a_n| < \epsilon$ e somar os primeiros n termos de S.

$$|a_n| < \epsilon \iff \frac{4}{2n+1} < \epsilon \iff 2n+1 > \frac{4}{\epsilon} \iff 2n > \frac{4}{\epsilon} - 1 \iff n > \frac{2}{\epsilon} - \frac{1}{2} \iff n \geq \frac{2}{\epsilon} \tag{6}$$

O programa utilizado foi:

```
def a(k):
    return 4*(-1)**k/(2*k+1)

s=0
indices=[2*10**i for i in range(8,16)]        #lista dos n para cada valor do erro
```

```
n=indices[7]+1                                    #num. max. de termos a somar
for k in range(n):
    if k in indices:
        print(k, s)
    s += a(k)
```

Obtêm-se os seguintes valores:

ϵ	n	S
10^{-8}	$2 * 10^{8}$	3.1415926485894077
10^{-9}	$2 * 10^{9}$	3.1415926530880767
10^{-10}	$2 * 10^{10}$	−
10^{-11}	$2 * 10^{11}$	−
10^{-12}	$2 * 10^{12}$	−
10^{-13}	$2 * 10^{13}$	−
10^{-14}	$2 * 10^{14}$	−
10^{-15}	$2 * 10^{15}$	−

CONCLUSÕES : n torna-se muito grande - o tempo de execuçao do programa cresce exponencialmente.

Exercício 4

Nos exercícios anteriores, o valor exato de S é π. Assim, podemos calcular o valor efetivo do erro absoluto, $E = |\pi - S|$, para cada valor aproximado de S.

Tomamos como valor exato $\pi = 3.1415926535897932$ (como tem 16 casas decimais corretas, o erro absoluto é sempre menor que $5 * 10^{-17}$ e por isso é menor que todos os ϵ)

TABELAS
CONCLUSÕES

Conclusão