

---

# TRABALHO 1

---

March 9, 2020

Análise Numérica (M2018)

Francisco Gonçalves  
201604505

Departamento de Ciência de Computadores  
Faculdade de Ciências de Universidade do Porto

## 0.1 PRIMEIRO EXERCÍCIO

### 0.1.1 Erro absoluto 1

Escrevam um programa que permita calcular um valor aproximado de

$$S = 18 \sum_{k=1}^{\infty} \frac{k!^2}{k^2(2k)!} \quad (1)$$

com erro absoluto inferior a  $\epsilon$  dado. O vosso programa deve imprimir o número  $n$  de termos somados na série e o valor aproximado de  $S$ . Usem o vosso programa para calcular valores aproximados de  $S$  com erro absoluto inferior a  $\epsilon = 10^{-8}, 10^{-9}, \dots, 10^{-15}$ .

**Listing 1:** Programa (PYTHON)

---

```
import math

for i in range (-8, -16, -1):
    count = 0
    k = 1
    sum = 0

    while (True):
        x = abs(math.factorial(k)**2 / (k**2 * math.factorial(2 * k)))
        if (x >= 10**i):
            sum += x
            count = count + 1
            k = k + 1
        else:
            break

    y = 18 * sum
    print('Erro =', 10**i, '| Numero de termos somados da serie:', count, '| Valor
        aproximado de S =', '%.16f' % y)
```

---

O programa calcula o valor de  $k$  de forma a que o erro absoluto não ultrapasse o valor de  $\epsilon$  em causa.

O valor da variável  $x$  representa cada valor da expressão  $\frac{k!^2}{k^2(2k)!}$  desde  $k = 1$  até o valor encontrado. O ciclo pára quando o valor de  $x$  for inferior ao erro, ou seja,  $10^i$ , sendo que  $i \in [-15, 8]$  e vai percorrendo esse intervalo por cada interação do ciclo.

Posteriormente, basta multiplicar o valor do somatório que está representado através da variável `sum` por 18 e imprimir a resposta desejada.

O número de termos somados da série em cada iteração é contado através de variável `count` que é incrementada sempre que, após um cálculo de  $x$  não é inferior ao erro especificado.

### 0.1.2 Resultados

⚠️NOTA: precisão usada neste exercício foi de 16 casas decimais

Erro	Número de termos somados	Valor aproximado de S
$10^{-8}$	11	9.869604342
$10^{-9}$	12	9.8696043878
$10^{-10}$	14	9.86960440042
$10^{-11}$	15	9.869604400938
$10^{-12}$	17	9.8696044010814
$10^{-13}$	18	9.86960440108752
$10^{-14}$	20	9.869604401089258
$10^{-15}$	21	9.8696044010893349

## 0.2 SEGUNDO EXERCÍCIO

### 0.2.1 Erro absoluto 2

Escrevam um programa que permita calcular um valor aproximado de

$$S = 12 \sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k^2} \quad (2)$$

com erro absoluto inferior a  $\epsilon$  dado. O vosso programa deve imprimir o número  $n$  de termos somados na série e o valor aproximado de  $S$ . Usem o vosso programa para calcular valores aproximados de  $S$  com erro absoluto inferior a  $\epsilon = 10^{-8}, 10^{-9}, \dots, 10^{-15}$ .

**Listing 2:** Programa (PYTHON)

---

```
import math

for i in range (-8, -16, -1):
    count = 0
    k = 1
    sum = 0

    while (True):
        x = ((-1)**(k-1)) / (k**2)
        sum = sum + x
        if (abs(x) >= 10**i):
            count = count + 1
            k = k + 1
        else:
            break

    y = 12 * sum
    print('Erro =', 10**i, '| Numero de termos somados da serie:', count, '| Valor
        aproximado de S =', '%.32f' % y)
```

---

Tal como no exercício anterior, o programa calcula o valor de  $k$  de forma a que o erro absoluto não ultrapasse o valor de  $\epsilon$  em causa.

O valor da variável  $x$  representa cada valor da expressão  $\frac{(-1)^{k-1}}{k^2}$  desde  $k = 1$  até o valor encontrado. O ciclo pára quando o valor de  $x$  for inferior ao erro, ou seja,  $10^i$ , sendo que  $i \in [-15, 8]$  e vai percorrendo esse intervalo por cada interação do ciclo.

Posteriormente, volta-se a multiplicar o valor do somatório que está representado através da variável `sum` por 12 desta vez e imprime-se a resposta desejada.

Tal como na implementação do exercício anterior, a variável `count` representa o número de termos somados da série.

### 0.2.2 Resultados

⚠️NOTA: precisão usada neste exercício foi de 32 casas decimais

Erro	Número de termos somados	Valor aproximado de S
$10^{-8}$	10000	9.8696044610
$10^{-9}$	31622	9.86960440708
$10^{-10}$	100000	9.869604401689
$10^{-11}$	316227	9.8696044010292
$10^{-12}$	1000000	9.86960440109500
$10^{-13}$	3162277	9.869604401088312
$10^{-14}$	10000000	9.8696044010890009
$10^{-15}$	31622776	9.86960440108894765

## 0.3 TERCEIRO EXERCÍCIO

### 0.3.1 Erro cometido 1

Sabendo que nos dois exercícios anteriores  $S = \pi^2$ , alterem os programas para imprimirem também o erro absoluto efetivamente cometido no cálculo de  $\pi^2$ ,  $E = |\pi^2 - S|$ . Comparem, interpretem e justifiquem os resultados.

**Listing 3:** Programa do exercício 1 (PYTHON)

---

```
import math

for i in range (-8, -16, -1):
    count = 0
    k = 1
    sum = 0

    while (True):
        x = abs(math.factorial(k)**2 / (k**2 * math.factorial(2 * k)))
        sum += x
        y = 18 * sum
        if (abs(math.pi**2 - y) >= 10**i):
            count = count + 1
            k = k + 1
        else:
            break

    z = abs(math.pi**2 - y)
    print('Erro =', 10**i, '| Numero de termos somados da serie:', count, '| Valor
        aproximado de S =', '%.16f' % z)
```

---

Após as modificações pedidas no exercício 3, o programa compara o valor da série para cada valor de  $k$  e verifica se o valor do erro absoluto é menor que o  $\epsilon$  escolhido, que corresponde a  $|\pi^2 - y|$ , visto que a variável  $y$  corresponde ao valor de  $S$  até ao valor de  $k$  da iteração atual.

Caso essa condição se verifique, o ciclo termina e imprime o erro absoluto calculado assim como o número de termos somados da série.

### 0.3.2 Resultados

⚠️ **NOTA:** precisão usada neste exercício foi de 16 casas decimais

Erro	Número de termos somados	Valor aproximado de S
$10^{-8}$	12	0.000000003
$10^{-9}$	13	0.0000000007
$10^{-10}$	15	0.00000000003
$10^{-11}$	16	0.000000000008
$10^{-12}$	18	0.0000000000004
$10^{-13}$	19	0.00000000000010
$10^{-14}$	21	0.000000000000005
$10^{-15}$	-	-

Não foi possível obter resultados para um erro de  $\epsilon = 10^{-15}$  devido a restrições de tempo ao correr o programa.

### 0.3.3 Erro cometido 2

**Listing 4:** Programa do exercício 2 (PYTHON)

---

```
import math

for i in range (-8, -16, -1):
    count = 0
    k = 1
    sum = 0

    while (True):
        x = ((-1)**(k-1)) / (k**2)
        sum = sum + x
        y = 12 * sum
        if (abs(math.pi**2 - abs(y)) >= 10**i):
            count = count + 1
            k = k + 1
        else:
            break

    z = abs(math.pi**2 - y)
    print('Erro =', 10**i, '| Numero de termos somados da serie:', count, '| Valor
        aproximado de S =', '%.16f' % z)
```

---

O programa para o exercício 2 sofreu exatamente as mesmas modificações que o exercício anterior logo a explicação é a mesma.



### 0.3.4 Resultados

⚠️ **NOTA:** precisão usada neste exercício foi de 16 casas decimais

Erro	Número de termos somados	Valor aproximado de S
$10^{-8}$	24494	0.000000010
$10^{-9}$	77450	0.0000000010
$10^{-10}$	244692	0.00000000010
$10^{-11}$	760944	0.000000000010
$10^{-12}$	2039462	0.0000000000010
$10^{-13}$	3323402	0.00000000000010
$10^{-14}$	3690030	0.000000000000009
$10^{-15}$	3718242	0.0000000000000000

Os programas aqui exibidos assim como os resultados obtidos podem ser consultados em detalhe aqui: <https://github.com/1Skkar1/Numerical-Analysis/tree/master/Trabalho1>