

# Tarefa 1

Kaio Henrique de Sousa

March 10, 2020

## Primeira Questão

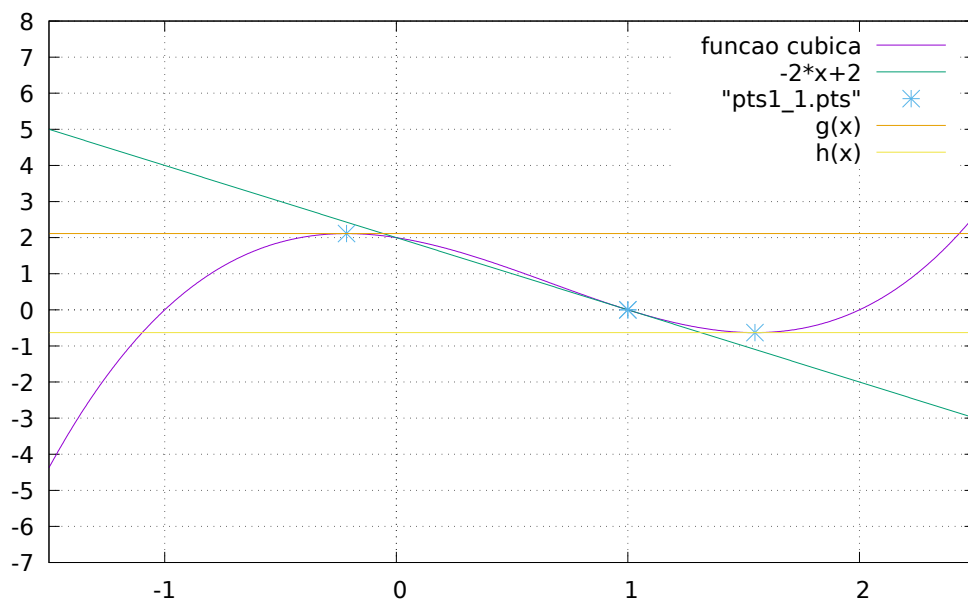


Figure 1: Plot da  $f(x)$  e os quatro pontos pedidos

## Segunda Questão

Primeiramente, foi utilizado a seguinte definição de derivada

$$f'(x) = \lim_{x \rightarrow a} \frac{f(x) - f(a)}{x - a} \quad (1)$$

depois, ao ignorar o limite e isolar  $f(a)$  teremos a seguinte equação

$$f(a) = f(x) - f'(a) * (x - a) \quad (2)$$

Agora, para cada ponto que vai ser calculado precisamos de um ponto  $x$  próximo ao  $a$ . Como o único ponto conhecido é o ponto  $f(0) = 0$ , iremos estimar cada ponto no intervalo  $[-6 : 6]$  a partir dele, que é exatamente o que ocorre no código do apêndice A, após os calculos é gerado um arquivos com os pontos da função no intervalo dado. Finalmente, temos o seguinte gráfico.

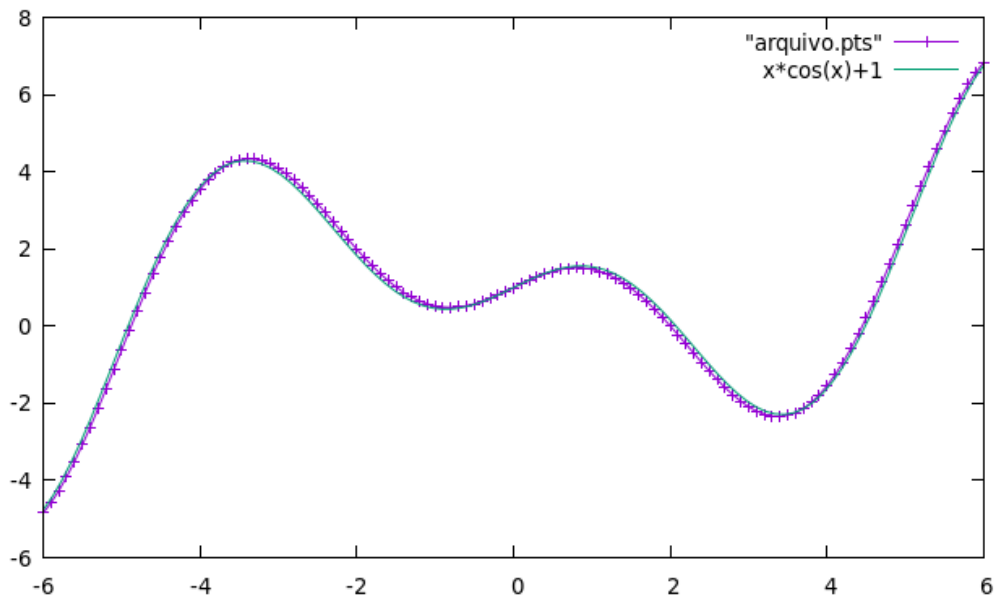


Figure 2: Gráfico da aproximação

Já na série de Taylor a seguir

$$f(x) = \sum_0^n \frac{f^n(a)}{n!} (x - a)^n \quad (3)$$

Como mostra no Apêndice B, foi definido a função fatorial e usando operador ternário foi definida a derivada  $f^{(n)}(x)$  para  $n$  ímpar ou par. Depois, foi definido a série segundo a equação (3), gerando o seguinte gráfico.

## Apêndice A

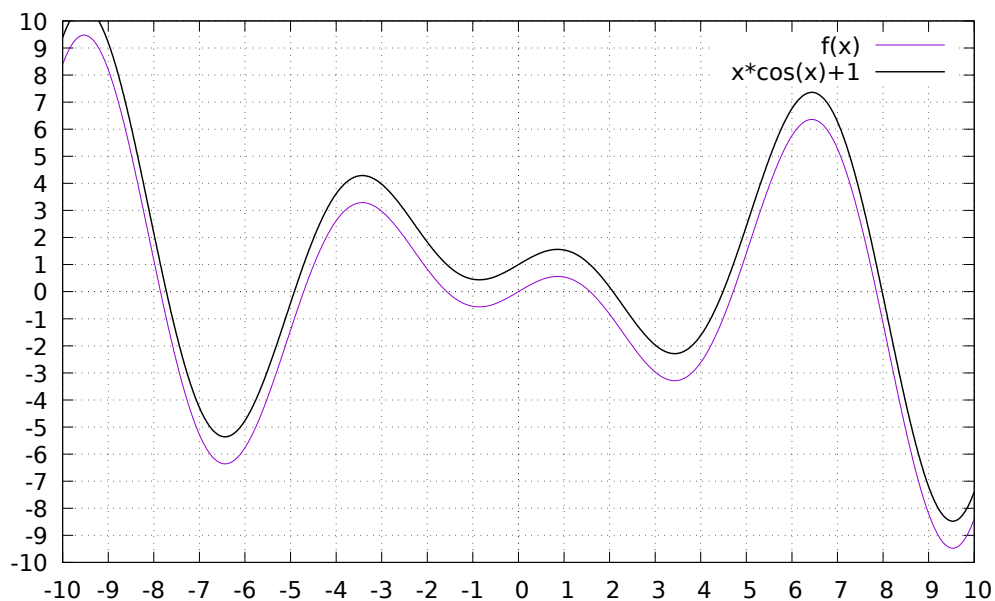


Figure 3: Série de Taylor com  $n=100$

Listing 1: Código em Python que estima a função desconhecida

---

```

1  import math
2
3  def hf(x):
4      return math.cos(x)-x*math.sin(x)
5
6  def f(x, y, a):
7      return y-hf(a)*(x-a)
8
9  alpha = 0
10 beta = 0
11
12 arquivo = open("arquivo.pts", 'w')
13
14 x_anterior = 0
15 y_anterior = 1
16 pontos_negativos = []
17
18 #Calculo dos pontos no intervalo [-6:0] a partir do ←
   ponto ponto (0, f(x)).
19 while(beta >= -6):
20     x_anterior = beta+0.1

```

```

21
22     if(beta == 0):
23         print("PONTO CONHECIDO")
24         pontos_negativos.append(str(beta)+ " " + str(f(↵
                (0, 1, beta)) + "\n")
25         y_anterior = f(0, 1, beta)
26     else:
27         pontos_negativos.append(str(beta) + " " + str(↵
                f(x_anterior, y_anterior, beta)) + "\n")
28         y_anterior = f(x_anterior, y_anterior, beta)
29
30     beta -= 0.1
31
32 #Escrita em ordem crescente
33 for i in reversed(pontos_negativos):
34     arquivo.write(i)
35
36 x_anterior = 0
37 y_anterior = 1
38
39 #Calculo dos pontos no intervalo [0:6] a partir do ↵
    ponto ponto (0, f(x)).
40 while(alpha <= 6):
41     x_anterior = alpha-0.1
42
43     if(alpha == 0):
44         print("PONTO CONHECIDO")
45         y_anterior = f(0, 1, alpha)
46     else:
47         arquivo.write(str(alpha) + " " + str(f(↵
                x_anterior, y_anterior, alpha)) + "\n")
48         y_anterior = f(x_anterior, y_anterior, alpha)
49
50     alpha += 0.1
51
52 arquivo.close()

```

---

## Apêndice B

Listing 2: Script que gera o gráfico da série de Taylor

---

```
1 set xrange [-10:10]
2 set yrange [-10:10]
3 set xtics 1
4 set ytics 1
5 set samples 1000
6 set style fill pattern 5
7 set xzeroaxis
8 set yzeroaxis
9 set grid
10
11 fac(n) = (n==0) ? 1 : n*fac(n-1)
12
13 hf(x, i) = (i%2 != 0) ? (-1)**((i-1)/2)*i*cos(x) + ↵
      (-1)**((i+1)/2)*x*sin(x) : (-1)**(i/2)*i*sin(x) + ↵
      (-1)**(i/2)*x*cos(x)
14
15 f(x) = sum [i=0:100] hf(0, i)*(x**i)/fac(i)
16 plot f(x) title "f(x)", x*cos(x)+1 lw 2 lc 0
```

---