



Universidade dos Açores

Cálculo II

Projeto Derivadas

Relatório de Projeto

Junho de 2020

Docente:

Prof. Doutor Paulo Medeiros

Realizado por:

Hélder F. M. de M. Braga

Índice

Introdução.....	6
Ferramentas Utilizadas.....	7
Funcionalidades.....	9
Código.....	18
Possíveis Melhorias.....	20
Conclusão.....	21
Webgrafia.....	22

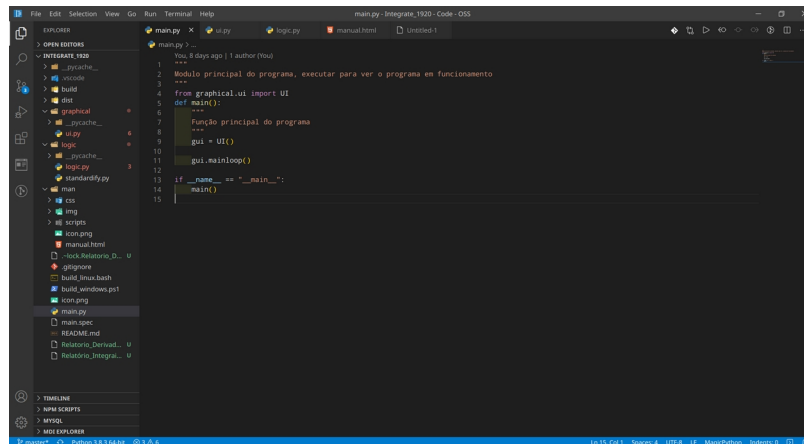
Introdução

Com a realização deste projeto pretendeu-se desenvolver um programa informático na linguagem de programação *Python* capaz de resolver alguns dos problemas relativos ao ramo do calculo de derivadas da disciplina de Calculo II. Além desta aplicação, foi também criado um texto histórico relativo à descoberta das metodologias do calculo diferencial e problemas de derivação.

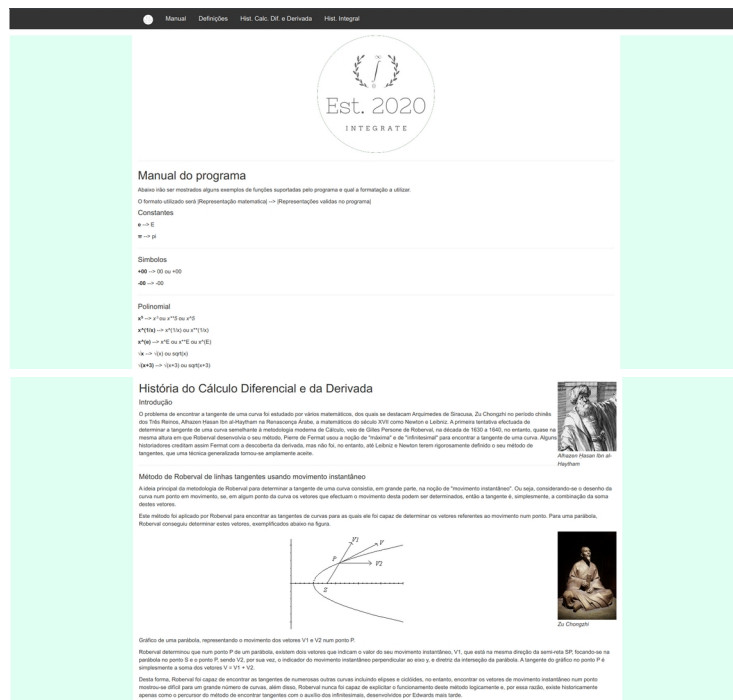
Ferramentas Utilizadas

Quanto as ferramentas utilizadas, temos as seguintes:

- Programa IDE: *Microsoft Code*



- Texto Histórico e Manual: *Bootstrap, CSS, HTML, Javascript*



- Programação:
 - Bootstrap
 - CSS
 - JavaScript
 - HTML
 - *Python 3.8.3*
 - Modulos:
 - *Matplotlib* → Criação de gráficos
 - *Numpy* → Lógica matemática
 - *Sympy* → Lógica matemática calculo diferencial
 - *PyInstaller* → Criação de executavel da aplicação
 - *Tkinter* → Interface gráfica
- Sistema(s) Operativo(s):
 - *Windows 10*
 - *Linux*
- Criação de executáveis:
 - *Modulo PyInstaller*
- Criação Setup:
 - *Nullsoft Scriptable Install System (NSIS)*
- Controlo de versões:
 - *Git*
 - *Github*

Funcionalidades

As funcionalidades implementadas no programa permitem proceder à demonstração algébrica e gráfica vários tipos de funções assim como derivação das mesmas facilitando assim a sua compreensão.

Pontos Desenvolvidos

- Mostrar o gráfico de uma função $f(x)$
- Mostrar a reta tangente de $f(x)$ num determinado ponto
- Calculo e representação da primeira derivada $f'(x)$ de uma função $f(x)$
- Mostrar o gráfico da primeira derivada $f'(x)$ de uma função
- Mostrar a reta tangente da primeira derivada $f'(x)$ de $f(x)$ num determinado ponto
- Calculo e representação da segunda derivada $f''(x)$ de uma função $f(x)$
- Mostrar o gráfico da segunda derivada $f''(x)$ de uma função $f(x)$
- Mostrar a reta tangente da segunda derivada de $f''(x)$ num determinado ponto
- Interface de utilizador gráfica com teclado
- Introdução de funções em modo de texto pelo utilizador
- Manipulação/customização de gráficos pelo utilizador
- Resumo da história da derivação e do calculo diferencial em formato de página web
- Manual de programa em formato web

Exemplos de uso

Função $f(x)$

Uma das funcionalidades do programa é mostrar o gráfico de uma função introduzida, como podemos ver na imagem com o exemplo da função $f(x)=x^3$.

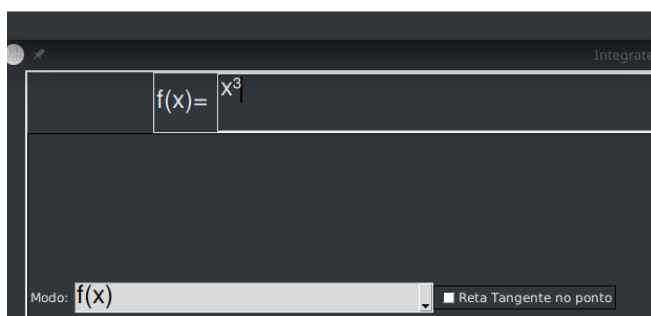
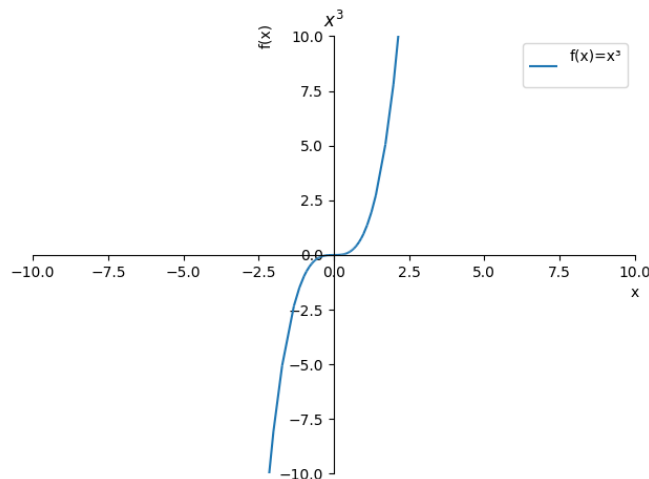


Figure 1: Gráfico da função x^3

A escala pode ser ajustada caso necessário, por exemplo, se quisermos outra “view” da mesma função, para tal basta definir as coordenadas mínimas e máximas de x e y como demonstrado na imagem.

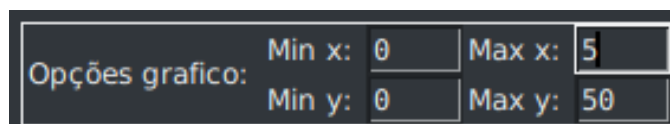


Figure 2: Opções do gráfico

Ficamos assim com a mesma função anterior, mas agora “cortada” para os valores de “0” a “5” para as abcissas e “0” a “50” para ordenadas

Além do ajuste da “viewport”, também é possível traçar a reta tangente num ponto da função, para tal, basta introduzir os valores do ponto na reta.

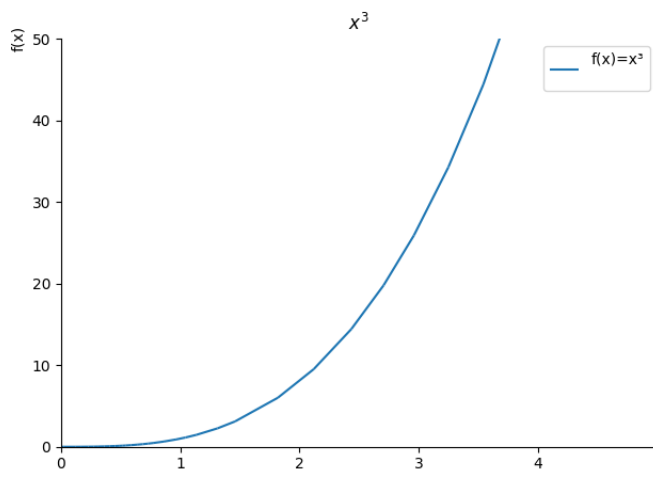


Figure 3: Gráfico função x^2 com “viewport” customizada

Obtemos agora um dos pontos da função como demonstrado na imagem. Neste exemplo, usaremos o ponto $P(2.005, 8.465)$.

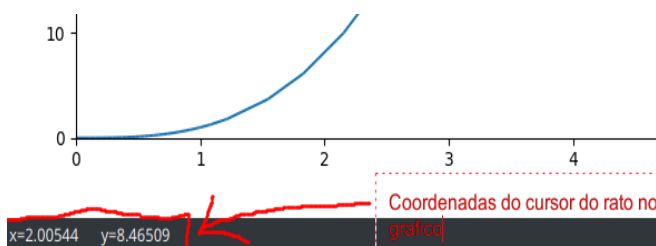


Figure 4: Coordenadas de um dos pontos de x^3

Introduzimos agora o ponto no campo relativo a reta tangente como pode ser visto a direita.

	x: 2.005
<input checked="" type="checkbox"/> Reta Tangente no ponto	y: 8.465
Margem erro:	

Figure 5: Opções Reta Tangente

O que nos dará a seguinte representação gráfica...

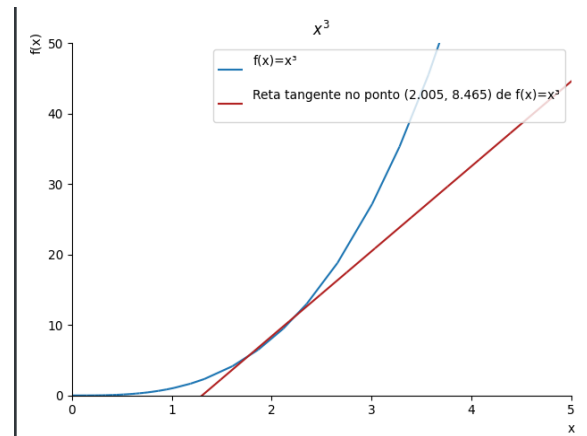


Figure 6: Gráfico da função x^3 com reta tangente num ponto

A margem de erro por predefinição é de 0.05, no entanto, podemos ajustá-la caso se necessário, quanto maior o valor introduzido na margem de erro, menos precisa será a reta tangente demonstrada no gráfico, no entanto, isto permite que no caso de existirem pontos não inteiros, por exemplo, $P(2.00544, 8.4509)$, o utilizador possa introduzir um ponto aproximado, como, $P(2.005, 8.4509)$ ou $P(2.0, 8.4)$ que terá uma representação gráfica semelhante, sem ser necessário uma precisão ao nível das milésimas.

Caso queira-mos grande precisão relativamente ao ponto para a reta tangente, podemos fazer como demonstrado na figura, no entanto, se o ponto não estiver dentro da margem de erro, surge uma mensagem de erro.

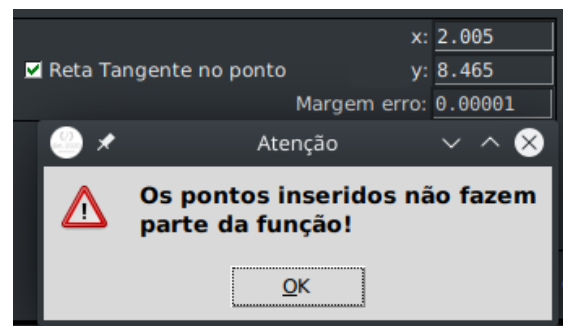


Figure 7: Reta tangente com grande precisão

Alternativamente, se precisão do ponto da reta tangente não for crucial, podemos por exemplo introduzir.

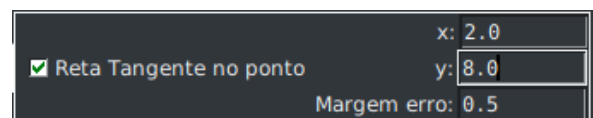


Figure 8: Reta tangente com pouca precisão

O que nos dará no gráfico...

Neste caso específico, a diferença entre o caso da “Figura 6” e da “Figura 9” é mínima, mas, tal pode nem sempre ser verdade, pelo que é aconselhável manter a margem de erro o mais pequena possível para evitar resultados incorretos.

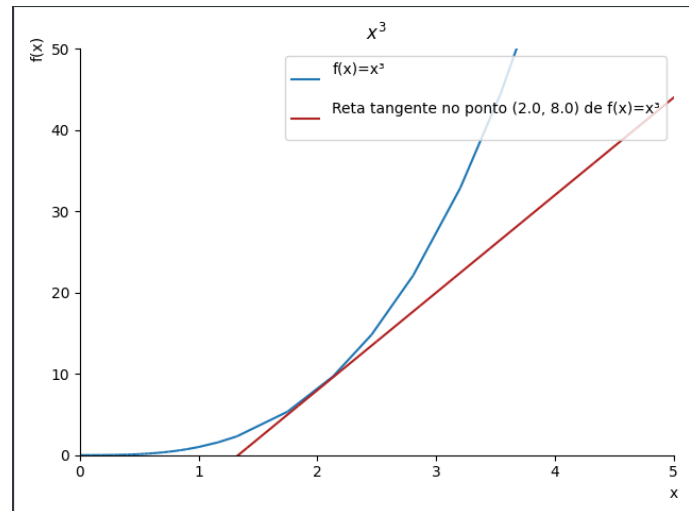


Figure 9: Reta tangente no ponto $P(2, 8)$

Um exemplo “extremo” de uma margem de erro demasiado generosa pode ser visto na “Figura 10”, o que nos dá resultados erróneos.

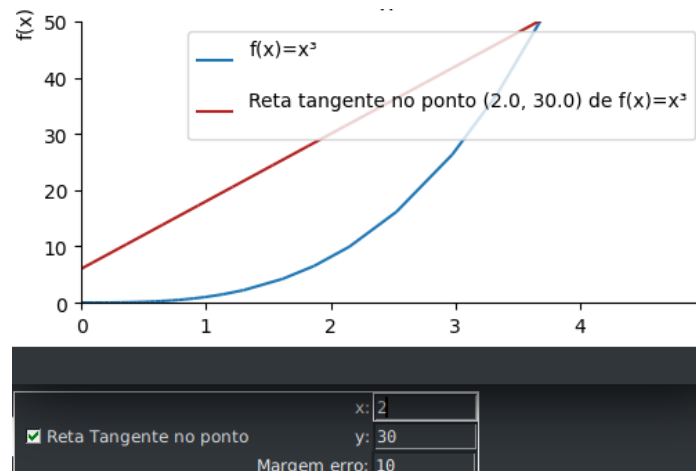


Figure 10: Reta tangente com margem de erro exagerada

Agora que já mostramos a manipulação de diversos aspetos dos gráficos da aplicação assim como expandi-mos no funcionamento da funcionalidade da tangente, passemos à primeira derivada da função x^3 .

Função $f'(x)$

Para obter a primeira derivada da função, começa-mos por seleccionar o modo “ $f'(x)$ ” no programa.

Ao clicar agora no botão “Resolver” o programa irá calcular a derivada de “ x^3 ” e mostrar o gráfico da mesma.

Como podemos ver na “*Figura 12*”, a função derivada de “ x^3 ” é “ $3x^2$ ”.

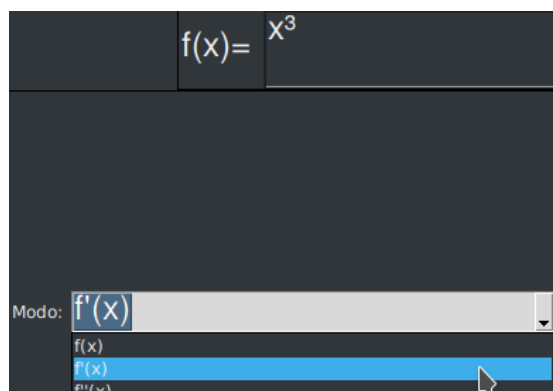


Figure 11: Modo $f'(x)$

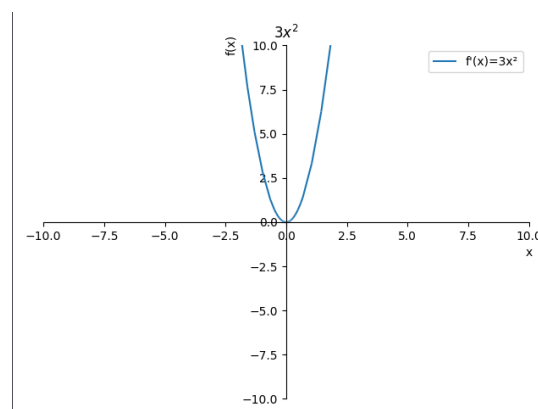


Figure 12: $f'(x)$ da função x^3

Caso queira-mos novamente a reta tangente num ponto para a função derivada, repetimos o processo que efetuamos na secção da função “ $f(x)$ ”.

Usando o ponto $P(1.295, 5.004)$ como exemplo obtemos a figura...

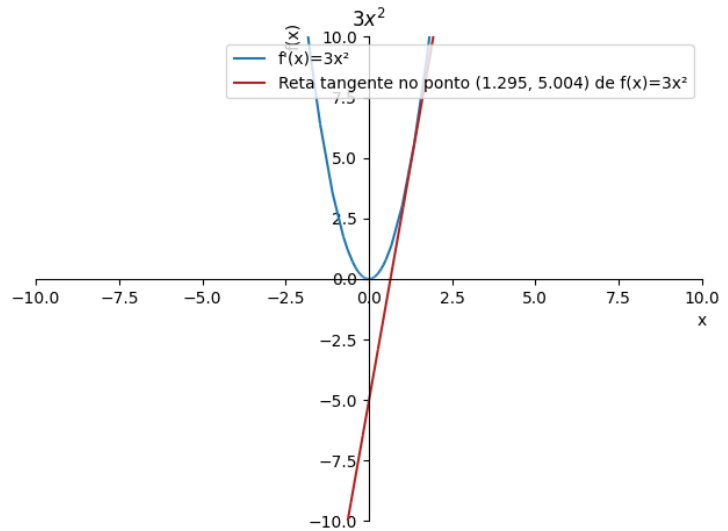


Figure 13: Função derivada com reta tangente

Função $f''(x)$

Para obter a segunda derivada, procedemos de forma semelhante ao que fizemos anteriormente no caso da primeira derivada.

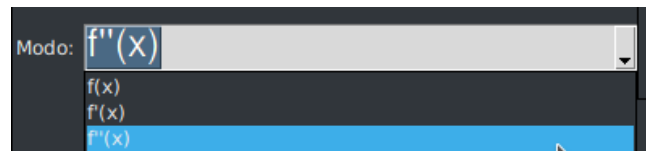


Figure 14: Modo segunda derivada $f''(x)$

Novamente, observado o gráfico obtido, vemos que a segunda derivada de $f(x) = x^3$ é igual a $f''(x) = 6x$

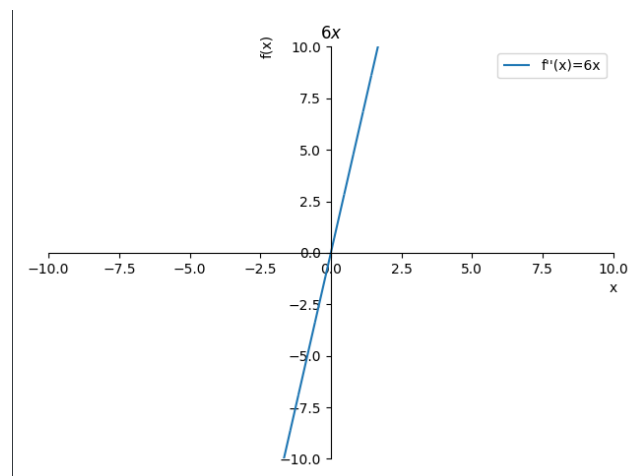


Figure 15: $f''(x)$ de x^3

Uma vez mais, usando a funcionalidade de reta tangente, com o ponto P(0.270, 1.626) como exemplo obtemos.

Como podemos observar, a reta da função e a sua reta tangente estão sobrepostas, pois a segunda derivada corresponde a uma função do primeiro grau.

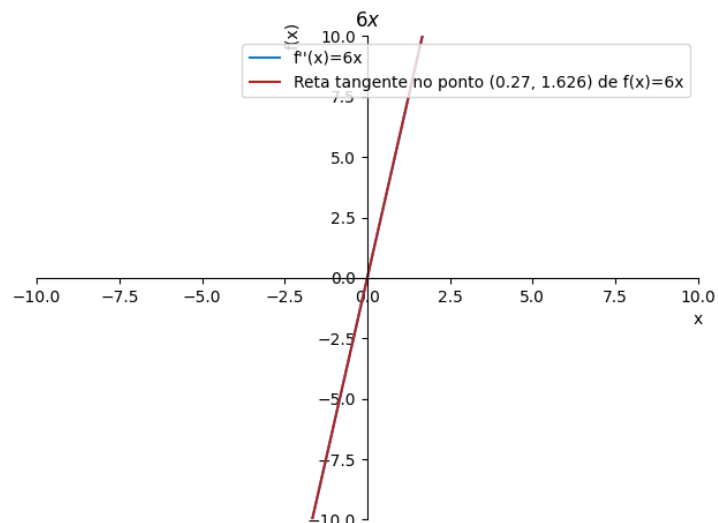


Figure 16: Reta tangente no ponto em $f'(x)$

No entanto, usando por exemplo a função

$f(x) =$

$$\frac{x^2}{\sqrt{x} + 3}$$

e obtendo a sua segunda derivada

$f''(x) =$

$$-\frac{7\sqrt{x}}{4(\sqrt{x} + 3)^2} + \frac{x}{2(\sqrt{x} + 3)^2} + \frac{2}{\sqrt{x} + 3}$$

obtemos a reta tangente no ponto P(0.1013, 0.4619) como demonstrado.

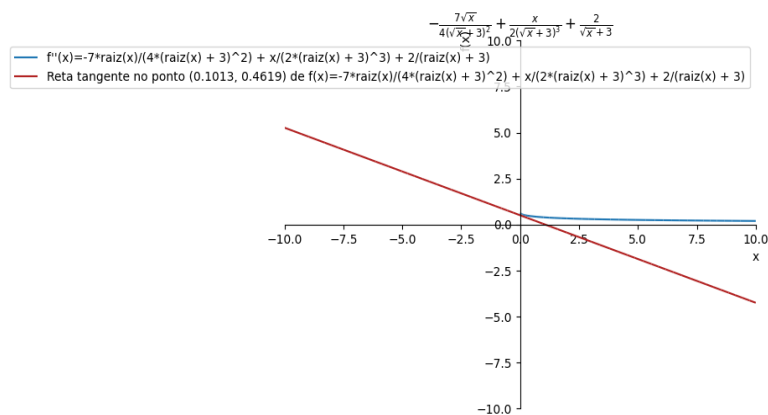


Figure 17: Exemplo alternativo de reta tangente no ponto em $f''(x)$

A introdução de funções no programa por sua vez pode ser feita de duas formas, utilizando os botões fornecidos na própria aplicação como demonstrado na imagem.

Operações:	+	-	*	/
	sin(x)	cos(x)	tan(x)	cotg(x)
	x^	√	()
	x	√(a,x)	log(x)	log_a(x)
	e	e^	ln(x)	π
	sec(x)	cosec(x)	arcsen(x)	arccos(x)
	arctan(x)	arccotan(x)	arcsec(x)	arccosec(x)

Figure 18: Painel de inserção de operações comuns da aplicação

Alternativamente, podemos introduzir diretamente a função desejada utilizando sintaxe do programa, por exemplo temos a seguinte função.

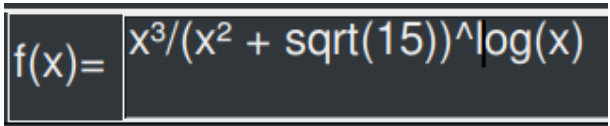

$$f(x) = x^3 / (x^2 + \sqrt{15})^{\log(x)}$$

Figure 19: Caixa de inserção da função

Quanto aos restante elementos, temos o botão “Apagar” que elimina o texto da função.

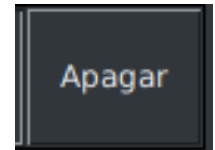


Figure 20:
Botão Apagar

Finalmente, temos o botão “Resolver”, que executa as operações selecionadas na aplicação, “Info” que mostra-nos o manual e a história do calculo diferencial e derivadas e o botão “Sair” que fecha a aplicação.

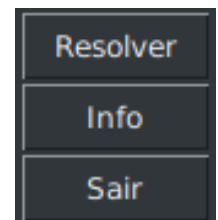


Figure 21:
Botões
execução
aplicação

Código

Explicação Geral

Não sendo a explicação de código foco do projeto, serão abordados apenas pontos gerais de secções relevantes, se for necessária alguma clarificação, pode ser consultado o “*docstring*” de cada função no código.

```
73 def inFuncao(self, funcao, coord_x, coord_y, margem_erro=0.05):
74     """Verifica se um ponto inserido faz parte de uma função tendo em conta uma margem de erro dada
75
76     Args:
77         funcao (string): Função a avaliar
78         coord_x (float): Coordenada x da reta tangente a função
79         coord_y (float): Coordenada y da reta tangente a função
80         margem_erro (float, optional): #Margem de erro para considerar um ponto dentro da função. Defaults to 0.00001.
81
82     Returns:
83         [boolean]: True se faz parte da função, False caso não faça
84     """
85     #Substitui x por valor da coordenada x
86     func_point = funcao.replace('x', "("+str(coord_x)+")")
87
88     #Resolve a função
89     func_point = s.sympify(func_point)
90
91     #Verifica se o ponto está na reta
92     return (coord_y*(1-margem_erro) < func_point and func_point < coord_y*(1+margem_erro))
```

Código que verifica se um determinado ponto faz ou não parte de uma função.

```
114 def DecliveValor(self, funcao, coord_x):
115     """Calcula o declive da reta num ponto de uma função recorrendo a derivada
116
117     Args:
118         funcao (string): Função para analisar
119         coord_x (float): Coordenada do ponto x
120
121     Returns:
122         float: declive
123     """
124     x = s.Symbol('x')
125     funcao_deriv = s.diff(funcao)
126
127     m = funcao_deriv.subs(x, coord_x)
128     return m
```

Calcula o declive da reta num determinado ponto dada a abcissa deste ponto e a função.

```

130 def retaTangentePonto(self, funcao, coord_x=0.0, coord_y=0.0, show_tangente=False, legenda="f(x)=", plot_eixo_x_limites=(-10, 10), plot_eixo_y_limites=(-10, 10)):
131     """Mostra o grafico de uma função, suporta mostrar a reta tangente num ponto (x, y) dadas as coordenadas
132
133     Args:
134         funcao (string): Função a mostrar em formato grafico
135         coord_x (float, optional): Ponto x da reta tangente. Defaults to 0.0.
136         coord_y (float, optional): Ponto y da reta tangente. Defaults to 0.0.
137         show_tangente (bool, optional): Mostrar reta tangente da função, True para mostrar, False para omitir. Defaults to False.
138         legenda (str, optional): Identificador do tipo de função, exemplos: f(x), f'(x), f''(x), etc. Defaults to "f(x)".
139     """
140
141     m, x, y = s.symbols('m x y')
142
143     funcao_pretty = self.handler.pretty_ready(funcao)
144
145     funcao_plot = s.plot(funcao, show=False)
146     funcao_plot.legend=True
147     funcao_plot[0].label=legenda+ funcao_pretty
148     funcao_plot.xlim = plot_eixo_x_limites
149     funcao_plot.ylim = plot_eixo_y_limites
150
151     #Reta tangente no ponto (x,y)
152     if show_tangente == True:
153         m = self.DecliveValor(funcao, coord_x)
154         y=s.sympify(m*(x-coord_x)+coord_y)
155
156         rt = s.plot(y, show=False)
157
158         rt.xlim = plot_eixo_x_limites
159         rt.ylim = plot_eixo_y_limites
160
161         funcao_plot.append(rt[0])
162
163         reta_tangente_pretty = "Reta tangente no ponto (" +str(coord_x)+", " +str(coord_y)+") de f(x)=" +self.handler.pretty_ready(funcao)
164         funcao_plot[1].label=reta_tangente_pretty
165         funcao_plot[1].line_color = 'firebrick'
166
167     funcao_latex = self.sympyLatexify(funcao)
168     funcao_plot.title = f"${funcao_latex}$"
169     funcao_plot.show()

```

Código responsável pelo gráfico da função e se selecionado, mostra também a reta tangente num ponto.

```

255 def derivate(self, funcao, ordem=1):
256     """Calcula a derivada de ordem n e retorna a sua expressao
257
258     Args:
259         funcao (string): Função a derivar
260         ordem (int, optional): Order da derivada, 1 para f'(x), 2 para f''(x), etc... Defaults to 1.
261
262     Returns:
263         [Derivative]: Expressao da derivada de ordem n
264     """
265     derivada_ordem_n = s.diff(funcao)
266     for i in range(ordem-1):
267         try:
268             if(str(derivada_ordem_n) == "0"):
269                 return derivada_ordem_n
270             derivada_ordem_n = s.diff(derivada_ordem_n)
271             derivada_ordem_n = str(derivada_ordem_n)
272
273         except Exception as e:
274             messagebox.showerror("Erro!",
275                                   "Não foi possivel calcular a derivada de ordem " + str(ordem) + " da função inserida!\n\n" + str(e))
276     return derivada_ordem_n

```

Função que efetua a derivada de ordem “n” de uma determinada função.

Possíveis Melhorias

Em termos de funcionalidade, a criação de um subsistema para permitir a atualização dos gráficos mostrados pela aplicação em tempo real (gerar o gráfico para novas coordenadas da “*viewport*” quando o utilizador o arrasta com o rato) poderia mostrar-se uma mais valia.

A implementação de uma interface gráfica que permita o cálculo de derivadas de ordem “ n ” poderia ser uma mais valia, caso se efetue uma análise da 3^o derivada ou 10^o derivada de uma função.

Conclusão

O projeto desenvolvido permite o cálculo, representação gráfica e algébrica de funções relativas cálculo diferencial, focando-se na derivação.

É possível, com a introdução de dados por parte do utilizador, uma melhor compreensão e análise de uma variedade de funções.

Associado ainda ao tema do cálculo diferencial, analisa-se o surgimento do cálculo de derivadas em termos históricos recorrendo a leitura do recurso em formato web, fornecido no programa.

Propõem-se também, potenciais melhorias que poderão ser implementadas, nomeadamente, derivadas de ordem “ n ” e a criação gráficos em tempo real.

Webgrafia

História Cálculo Diferencial e Derivada

<https://en.wikipedia.org/wiki/Derivative>

https://en.wikipedia.org/wiki/History_of_calculus

[https://en.wikipedia.org/wiki/Fermat%27s_theorem_\(stationary_points\)](https://en.wikipedia.org/wiki/Fermat%27s_theorem_(stationary_points))

<http://www.math.wpi.edu/IQP/BVCalcHist/calc2.htm>

https://en.wikipedia.org/wiki/Numerical_differentiation

https://en.wikipedia.org/wiki/Symmetric_derivative

Software

Code IDE

<https://code.visualstudio.com/>

Python

<https://www.python.org/>

Módulo Numpy

<https://numpy.org/>

Módulo Matplotlib

<https://matplotlib.org/>

Módulo PyInstaller

<https://www.pyinstaller.org/>

Módulo Scipy

<https://www.scipy.org/>

Módulo Sympy

<https://www.sympy.org/en/index.html>

NSIS

https://nsis.sourceforge.io/Main_Page

Repositório do projeto no Github:

https://github.com/hfmmb/Integrate_1920

Repositório do projeto no Github (Versão executável):

https://github.com/hfmmb/Integrate_1920/releases