Universidade dos Açores

Cálculo II

Projeto Integrais

Relatório de Projeto

Junho de 2020

Docente:

Prof. Doutor Paulo Medeiros

Realizado por:

Hélder F. M. de M. Braga

Índice

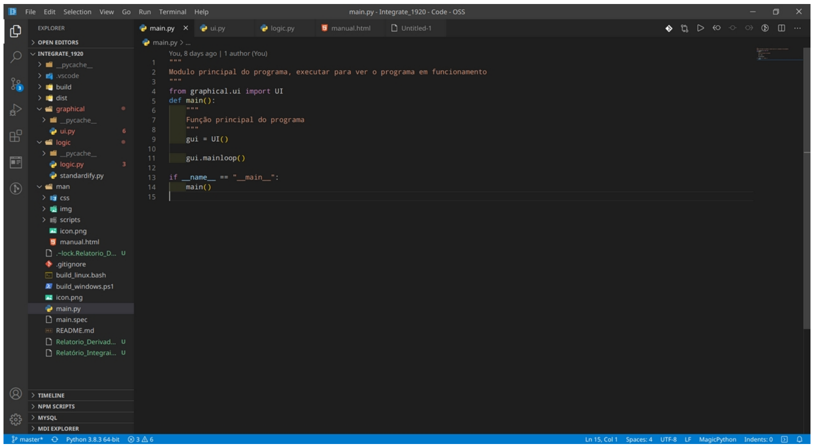
Introdução

Com a realização deste projeto pretendeu-se adicionar funcionalidades ao programa já desenvolvido no projeto das derivadas anterior, de modo a que este conseguisse também efetuar operações relativas ao calculo integral no contexto da disciplina de Calculo II. Além da expansão da aplicação, foram também adicionados elementos acerca da história da descoberta do calculo de integrais e o calculo de zeros, limites e mínimos e máximos de uma função.

Ferramentas Utilizadas

Visto tratar-se de uma expansão do projeto anterior, grande parte das ferramentas utilizadas repetem-se.

- 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

Como o projeto expande um projeto anterior já desenvolvido, serão referidas apenas as funcionalidades adicionadas nesta iteração.

Pontos Desenvolvidos

- Calcular e mostrar a expressão de um integral indefinida / primitiva de uma função dada.
- Calcular e mostrar a expressão de um integral definida.
- Calcular o valor aproximado de uma integral definida.
- Mostrar o gráfico de uma integral definida.
- Calcular e mostrar a expressão de um integral imprópria.
- Calcular o valor aproximado de uma integral imprópria.
- Calcular os mínimos e máximos de uma função $f(x)$ se existirem.
- Calcular o valor do limite de $f(x)$ quando este tende para um determinado valor.
- Calcular os zeros de uma função $f(x)$ caso existam.
- Expansão do manual do programa.
- História do cálculo diferencial com uma nova secção focada nas integrais.

Exemplos de uso

$\int f(x)dx$ **Indefinida**

Reaproveitando a função usada no relatório das derivadas para exemplificação, sendo esta *f(x)=x³*, colocaremos agora a função no programa.

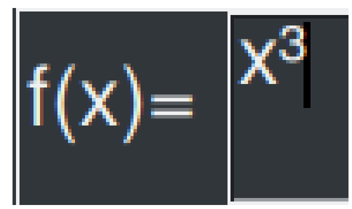


Figure 1: Função inserida

Feito isto, procede-mos a colocar o programa no modo de integral indefinida/ primitiva.

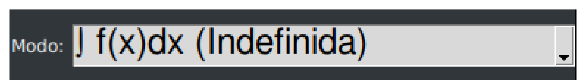


Figure 2: Modo Integrais Indefinidas / Primitivas

Terminado o cálculo, obtemos agora a seguinte expressão função primitiva como resultado.

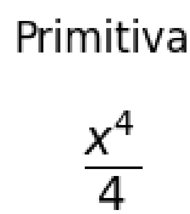


Figure 3: Primitiva Obtida

$\int_a^b f(x)dx$ **Definida**

Inseri-mos agora a primitiva obtida na aplicação.

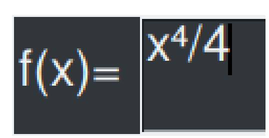


Figure 4: Função primitiva inserida7

Coloca-mos agora a aplicação em modo de integral definida.

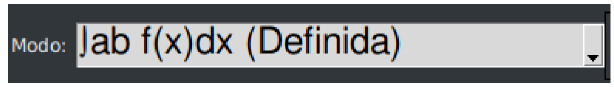


Figure 5: Modo integral definida

Procedemos agora a inserir os limites *"a"* e *"b"* da integral definida, neste caso, usare-mos os valores $a=1$ e $b=5$.

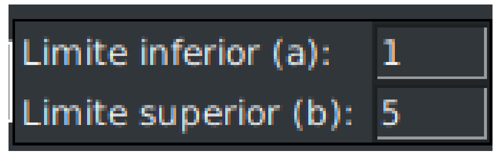


Figure 6: Limites integral definida

Ao executar o modo, obtemos a seguinte expressão para a integrada definida.

$$\int_1^5 f(x)\,dx = \frac{x^4}{4}$$

Figure 7: Expressão Integral definida

E o seguinte gráfico da integral assim como o valor aproximado da mesma calculado para os limites inseridos.

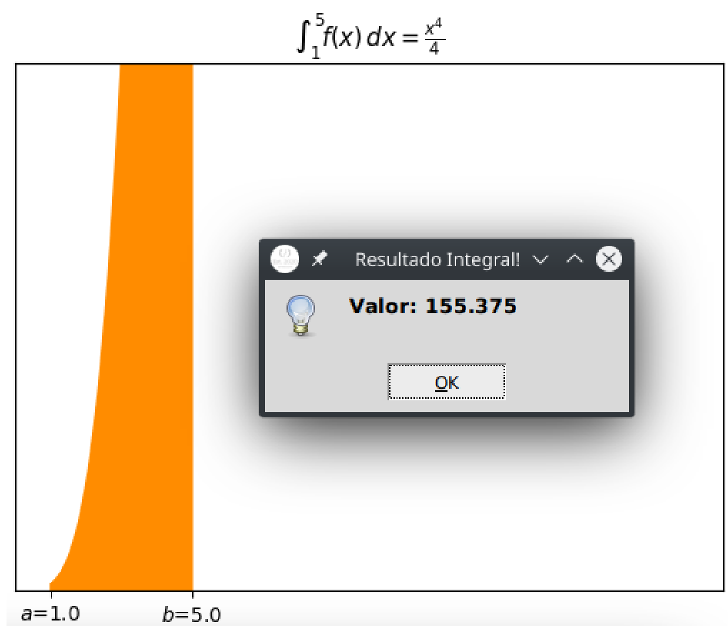


Figure 8: Gráfico Integral Definida

Utilizando o mesmo processo, mas com uma função mais complexa, obtemos o seguinte gráfico, expressão e valor aproximado

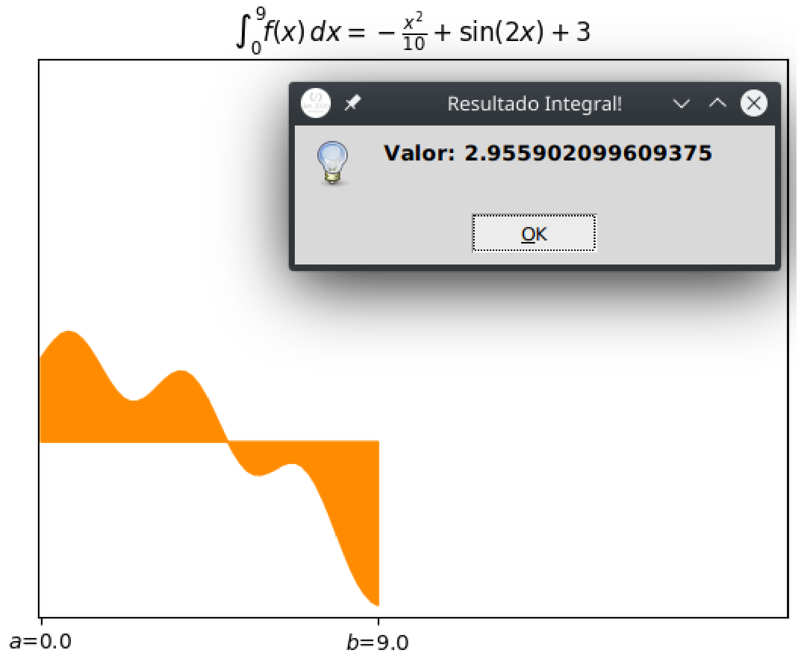


Figure 9: Gráfico integral definida função complexa

$\int_a^b f(x)dx$ **Imprópria**

Passando agora ao cálculo de integrais impróprias, utilizando a mesma função anterior f(x)= $\frac{x^4}{4}$ e definindo *a=5* e *b=+oo* obtemos o seguinte resultado.

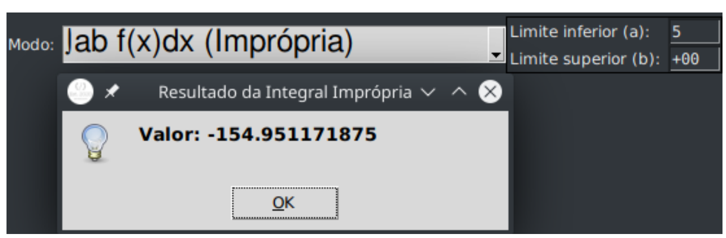


Figure 10: Calculo do valor aproximado de uma integral imprópria

Mínimos e Máximos de *f(x)*

Quanto a mínimos e máximos, obtemos o seguinte.

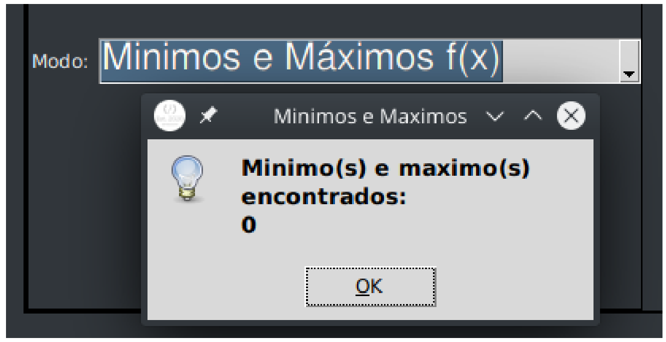


Figure 11: Mínimos e máximos de uma função

Valor Limite f(x)

Quanto aos limites, ao testar a tendência do limite para +00 para f(x) obtemos como resultado +00

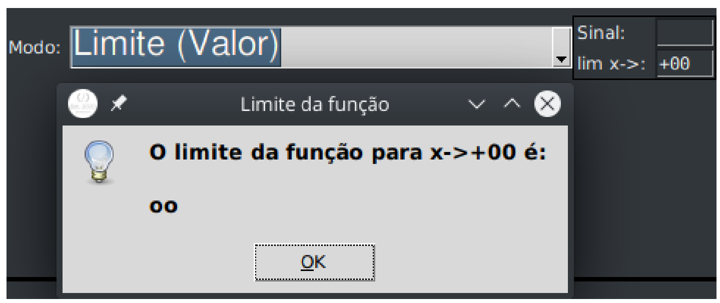


Figure 12: Valor do calculo de um limite a tender para +00

Outro caso possível de uso, é o demonstrado, neste caso, o limite de f(x) a tender para 2- obtemos o valor 4.

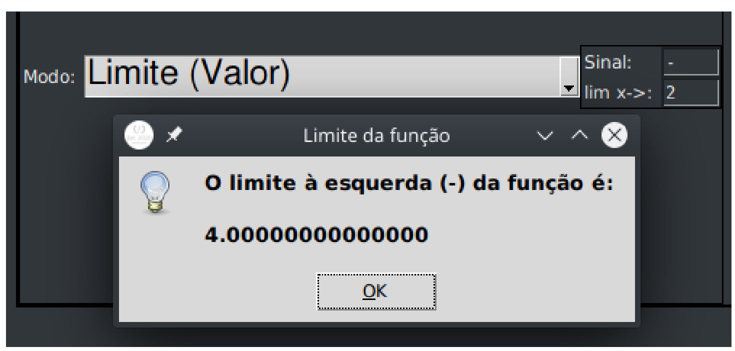


Figure 13: Valor do calculo de um limite a tender para 2 a esquerda da função

Zeros f(x)

Quanto aos zeros, a função tem apenas 1 zero sendo este 0.

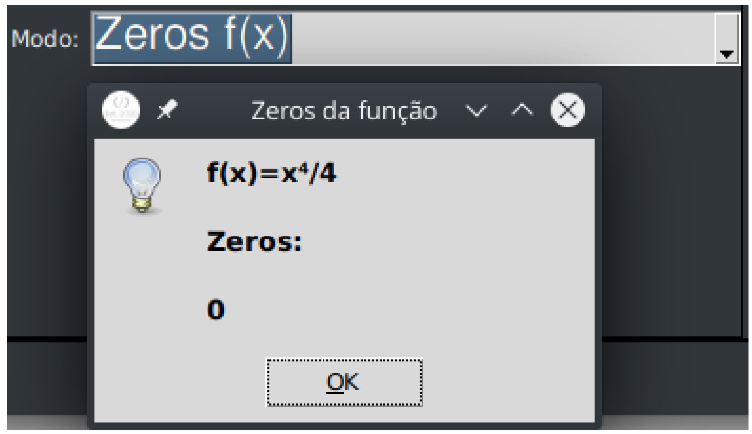
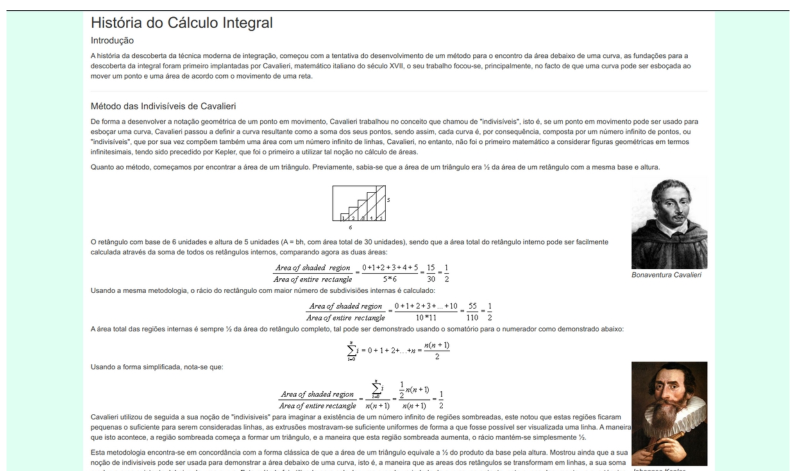


Figure 14: Zeros de uma função

História Integrais

Tendo terminado agora a explicação das funcionalidades adicionadas, em termos da história das integrais, foi adicionada a seguinte secção.



História do Cálculo Integral

Introdução

A história da descoberta da técnica moderna de integração, começou com a tentativa do desenvolvimento de um método para o encontro da área debaixo de uma curva, as fundações para a descoberta da integral foram primeiro implantadas por Cavalieri, matemático italiano do século XVII, o seu trabalho focou-se, principalmente, no facto de que uma curva pode ser esboçada ao mover um ponto e uma área de acordo com o movimento de uma reta.

Método das Indivisíveis de Cavalieri

De forma a desenvolver a notação geométrica de um ponto em movimento, Cavalieri trabalhou no conceito que chamou de "indivisíveis", isto é, se um ponto em movimento pode ser usado para esboçar uma curva, Cavalieri passou a definir a curva resultante como a soma dos seus pontos, sendo assim, cada curva é, por consequência, composta por um número infinito de pontos, ou "indivisíveis", que por sua vez compõem também uma área com um número infinito de linhas, Cavalieri, no entanto, não foi o primeiro matemático a considerar figuras geométricas em termos infinitesimais, tendo sido precedido por Kepler, que foi o primeiro a utilizar tal noção no cálculo de áreas.

Quanto ao método, começamos por encontrar a área de um triângulo. Previamente, sabia-se que a área de um triângulo era ½ da área de um retângulo com a mesma base e altura.

Bonaventura Cavalieri

O retângulo com base de 6 unidades e altura de 5 unidades (A = bh, com área total de 30 unidades), sendo que a área total do retângulo interno pode ser facilmente calculada através da soma de todos os retângulos internos, comparando agora as duas áreas:

$$\frac{\textit{Area of shaded region}}{\textit{Area of entire rectangle}} = \frac{0+1+2+3+4+5}{5*6} = \frac{15}{30} = \frac{1}{2}$$

Usando a mesma metodologia, o rácio do rectângulo com maior número de subdivisiões internas é calculado:

$$\frac{\textit{Area of shaded region}}{\textit{Area of entire rectangle}} = \frac{0+1+2+3+...+10}{10*11} = \frac{55}{110} = \frac{1}{2}$$

A área total das regiões internas é sempre ½ da área do retângulo completo, tal pode ser demonstrado usando o somatório para o numerador como demonstrado abaixo:

$$\sum_{i=0}^{n} i = 0+1+2+\ldots+n = \frac{n(n+1)}{2}$$

Usando a forma simplificada, nota-se que:

$$\frac{\textit{Area of shaded region}}{\textit{Area of entire rectangle}} = \frac{\sum_{i=0}^{n} i}{n(n+1)} = \frac{\frac{1}{2}n(n+1)}{n(n+1)} = \frac{1}{2}$$

Cavalieri utilizou de seguida a sua noção de "indivisiveis" para imaginar a existência de um número infinito de regiões sombreadas, este notou que estas regiões ficaram pequenas o suficiente para serem consideradas linhas, as extrusões mostravam-se suficiente uniformes de forma a que fosse possível ser visualizada uma linha. A maneira que isto acontece, a região sombreada começa a formar um triângulo, e a maneira que esta região sombreada aumenta, o rácio mantém-se simplesmente ½.

Esta metodologia encontra-se em concordância com a forma clássica de que a área de um triângulo equivale a ½ do produto da base pela altura. Mostrou ainda que a sua noção de indivisiveis pode ser usada para demonstrar a área debaixo de uma curva, isto é, a maneira que as areas dos retângulos se transformam em linhas, a sua soma produz a area existente debaixo de uma curva. Este método foi utilizado para calcular uma grande variedade de curvas, no entanto, este nunca documentou a sua técnica

Johannes Kepler

Código

Visto que a analise do código não é o foco deste projeto, irá ser feita apenas uma explicação generalizada de algumas das secções consideradas mais relevantes. No entanto, caso seja necessário um esclarecimento mais aprofundado, pode-se proceder à leitura das *"docstrings"* de cada função.

Explicação Geral

```python
def LimiteValor(self, funcao, valor_tendencia_limite, sinal=None, margem_erro=0.1):
    """Efectua o calculo de um limite e retorna o valor caso exista

    Args:
        funcao (string): Função a ser analizada para a existencia de limites
        valor_tendencia (float): Valor para qual o limite tende (lim f(x), x->y)
        sinal (string, optional): Limite a esquerda (-), direita (+) ou ambos (Campo vazio). Defaults to None.
        margem_erro (float): Valor de tolerancia usado para saber se o limite existe ou não, default é 0.1

    Returns:
        float: Retorna o valor do limite se este existe, None se este nao existe
    """

    x = s.Symbol('x')
    try:
        funcao = s.sympify(funcao)
    except s.SympifyError as e:
        print("Função em formato incorreto:\n", e)
    try:
        if(valor_tendencia_limite != "-00" and valor_tendencia_limite != "00" and valor_tendencia_limite != "+00"):
            valor_tendencia_limite = float(valor_tendencia_limite)

    except ValueError as e:
        print("Valor invalido: ", str(e))

    #Limite a esquerda da função
    if(sinal=="-"):
        if valor_tendencia_limite == "-00":
            return s.limit(funcao, x, s.S.NegativeInfinity, dir=sinal)

        elif valor_tendencia_limite == "00" or valor_tendencia_limite == "+00":
            return s.limit(funcao, x, s.S.Infinity, dir=sinal)

        elif(funcao.subs(x, valor_tendencia_limite-1e-8).is_real):
            return s.limit(funcao, x, valor_tendencia_limite, dir=sinal)
        else:
            return None #Limite não existe

    #Limite a direita da função
    elif(sinal=="+"):
        if valor_tendencia_limite == "-00":
            return s.limit(funcao, x, s.S.NegativeInfinity, dir=sinal)

        elif valor_tendencia_limite == "00" or valor_tendencia_limite == "+00":
            return s.limit(funcao, x, s.S.Infinity, dir=sinal)

        elif(funcao.subs(x, valor_tendencia_limite+1e-8).is_real):
            return s.limit(funcao, x, valor_tendencia_limite, dir=sinal)
        else:
            return None #Limite não existe
    else:
        if valor_tendencia_limite == "-00":
            return s.limit(funcao, x, s.S.NegativeInfinity)

        elif valor_tendencia_limite == "00" or valor_tendencia_limite == "+00":
            return s.limit(funcao, x, s.S.Infinity)
        else:
            sinal_esquerda=funcao.subs(x, str(valor_tendencia_limite-1e-8))
            sinal_direita=funcao.subs(x, str(valor_tendencia_limite+1e-8))
            #Assume-se que o limite seja um valor pequeno,
            #se este for pequeno o suficiente,
            #assumimos que este existe e tentamos calcula-lo
            if(abs(sinal_esquerda-sinal_direita)<margem_erro):
                return s.limit(funcao, x, valor_tendencia_limite)
            else:
                return None #Limite não existe
```

Secção do código responsável por calcular o valor da tendência de um limite de uma função dada caso este exista.

```python
    def Intergral_Valor(self, funcao, limite_inferior=None, limite_superior=None):
        """Retorna o valor aproximado da integral da função dada

        Args:
            funcao (string): Função a integrar
        """
        x = s.Symbol('x')
        integral = s.Integral(funcao, (x, limite_inferior, limite_superior))
        try:
            funcao = s.sympify(funcao)
        except s.SympifyError as err:
            return None
        try:
            integral = float(integral.as_sum(10).n(4))
            return integral
        except Exception as err:
            print(err)
        return None
```

Função responsável por calcular o valor aproximado de uma integral definida ou imprópria.

```python
def plotIntegralDefinida(self, funcao, a=1e-32 , b=None, plot_eixo_x_limites=(0, 20), plot_eixo_y_limites=(0, 20)):
    #Ajusta o eixo x minimo no grafico
    if(plot_eixo_x_limites[0] < 0):
        plot_eixo_x_limites[0] = 0
    #Ajusta o eixo y minimo no grafico
    if(plot_eixo_y_limites[0] < 0):
        plot_eixo_y_limites[0] = 0

    x = s.symbols("x")
    funcao = s.sympify(funcao)

    w = lambdify(x, funcao, "numpy")

    fig, aq = mplot.subplots()
    aq.set_ylim(bottom=0)

    # Make the shaded region
    iq = np.linspace(a, b)
    iw = w(iq)
    verts = [(a, 0), *zip(iq, iw), (b, 0)]
    poly = mt.patches.Polygon(verts, color='darkorange', edgecolor='black')
    aq.add_patch(poly)

    aq.text(0.5 * (a + b), 30, r"$\int_a^b f(q)\mathrm{d}q$",
            horizontalalignment='center', fontsize=20)

    aq.xaxis.set_ticks_position('bottom')

    aq.set_xticks((a, b))
    aq.set_xticklabels(('$a$'+'='+str(a), '$b$'+'='+str(b)))
    aq.set_yticks([])

    funcao_latex = self.sympyLatexify(funcao)

    mapa = {'a':str(int(a)), 'b':str(int(b)), 'f':funcao_latex}
    integral_string = r"\int_{a}^{b} \! f(x) \,dx= {f}".format_map(mapa)

    mplot.title(f"${integral_string}$")
    mplot.xlim(plot_eixo_x_limites[0], plot_eixo_x_limites[1])
    mplot.ylim(plot_eixo_y_limites[0], plot_eixo_y_limites[1])

    mplot.show()
```

Função responsável por mostrar graficamente uma integral definida limitada por um ponto *“a”* e *“b”*.

```html
<section id="historia_integral">
  <h2>História do Cálculo Integral</h2>
  <h4>Introdução</h4>
  <p>A história da descoberta da técnica moderna de integração, começou com a tentativa do desenvolvimento de um método para o encontro da área debaixo de uma curva, as
  <hr>
  <h4>Método das Indivisíveis de Cavalieri</h4>
  <p>De forma a desenvolver a notação geométrica de um ponto em movimento, Cavalieri trabalhou no conceito que chamou de "indivisíveis", isto é, se um ponto em movimento
  <p>Quanto ao método, começamos por encontrar a área de um triângulo. Previamente, sabia-se que a área de um triângulo era ½ da área de um retângulo com a mesma base e
  <div style="float:right; max-width: 150px; max-height: 250px;">
    <img src="img/people/bonaventuraCavalieri.jpeg" alt="Cavalieri" style="float: right; max-width: 150px; max-height: 250px;">
    <i style="vertical-align: middle;">Bonaventura Cavalieri</i>
  </div>
<img class="center" src="img/img_17.gif">
  <p>O retângulo com base de 6 unidades e altura de 5 unidades (A = bh, com área total de 30 unidades), sendo que a área total do retângulo interno pode ser facilmente c
  <img class="center" src="img/img_18.gif">
  <p>Usando a mesma metodologia, o rácio do rectângulo com maior número de subdivisiões internas é calculado:</p>
  <img class="center" src="img/img_19.gif">
  <p>A área total das regiões internas é sempre ½ da área do retângulo completo, tal pode ser demonstrado usando o somatório para o numerador como demonstrado abaixo:</p
  <img class="center" src="img/img_20.gif">
  <div style="float:right; max-width: 150px; max-height: 250px;">
    <img src="img/people/JohannesKepler.jpg" alt="Johannes Kepler" style="float: right; max-width: 150px; max-height: 250px;">
    <i style="vertical-align: middle;">Johannes Kepler</i>
  </div>
  <p>Usando a forma simplificada, nota-se que:</p>
  <img class="center" src="img/img_21.gif">
  <p>Cavalieri utilizou de seguida a sua noção de "indivisiveis" para imaginar a existência de um número infinito de regiões sombreadas, este notou que estas regiões fic
    A maneira que isto acontece, a região sombreada começa a formar um triângulo, e a maneira que esta região sombreada aumenta, o rácio mantém-se simplesmente ½.</p>
  <p>Esta metodologia encontra-se em concordância com a forma clássica de que a área de um triângulo equivale a ½ do produto da base pela altura. Mostrou ainda que a sua
  <p>De forma a entender as contribuições de Wallis para o cálculo integral, é necessario analisar primeiro as técnicas teoricas de Cavalieri aplicando-as para encontrar
  <img class="center" src="img/img_23.gif">
  <p>Cada região retangular tem base de 1 unidade relativamente ao eixo das abcissas e altura de x2 (obtida através da definição de parábola). Definindo o número de regi
  <img class="center" src="img/img_24.gif">
  <div style="float:right; max-width: 150px; max-height: 250px;">
    <img src="img/people/johnWallis.jpg" alt="Leibniz" style="float: right; max-width:150px; max-height: 250px;">
    <i style="vertical-align: middle;">John Wallis</i>
  </div>
  <p>Relembrando agora que a área de um retângulo é definida pelo produto da sua base e altura, pode ser afirmado que o retângulo encapsulante tem m+1 na sua base e m2 n
  <img class="center" src="img/img_25.gif">
  <p>Cavalieri procedeu a utilizar os seus "indivisíveis" para fazer outra descoberta no ramo do cálculo, e notou que à medida que m ficava maior, o termo <img src="img/
  <img class="center" src="img/img_27.gif">
  <p>Ou seja, de forma a que o número de retângulos tende para infinito, o rácio das áreas aproxima-se de <img src="img/img_29.gif">. Apesar disto, Cavalieri nunca formal
  <img class="center" src="img/img_31.gif">
  <p>Com este metodo, Cavalieri tomou os primeiros passos face ao desenvolvimento da integração.</p>
  <hr>
  <h4>Lei da Integração Polinomial de Wallis</h4>
  <p>A contribuição de John Wallis para o cálculo integral foi desenvolver uma lei algébrica de integração que reduziu a necessidade de analisar cada curva. Examinando a
  <p>Consideremos o gráfico da função y = k ou <img src="img/img_32.gif">:</p>
  <img class="center" src="img/img_33.gif">
  <p>Como podemos observar no diagrama, a área abaixo da linha em qualquer ponto ao longo das abcissas será kx ou <img src = "img/img_34.gif">. Consideremos agora o gráf
```

História das Integrais em formato web utilizando a linguagem HTML.

Possíveis Melhorias

Visto a integração tratar-se de uma operação matemática complexa, algumas das funções introduzidas podem provar-se computacionalmente dispendiosas, quer em termos de memória como em termos de tempo de processamento, sendo assim, uma das possíveis melhorias seria a paralelização de computações matemáticas com o uso de "*threads*". Outra alteração possível, é a implementação de algoritmos adicionais para a resolução de integrais *"exóticas"* não suportadas de momento.

Conclusão

O programa permite obter a expressão da primitiva, integral definida e impropria dada uma função, o cálculo e valor aproximado de integrais definidas ou impróprias e a representação gráfica de integral definidas. Permite ainda o calculo de zeros, valor de limites dada uma tendência e mínimos e máximos absolutos e relativos de uma função.

Adicionalmente, foi realizada a expansão do manual de utilizador e da secção histórica já existente, adicionando um segmento de calculo integral.

Demonstra-se exemplos de operações para clarificar o seu entendimento, procedendo-se a uma explicação de segmentos relevantes do código desenvolvido, procede-se ainda para à explicação de potenciais melhorias a implementar.

Webgrafia

História Cálculo Diferencial e Derivada

https://www.its.caltech.edu/~kcborder/Notes/LeibnizRule.pdf

https://mathcs.clarku.edu/~ma121/fermat.pdf

https://mindyourdecisions.com/blog/2016/10/12/the-wallis-product-formula-for-pi-and-its-proof/

http://www.math.wpi.edu/IQP/BVCalcHist/calc1.html

http://www.math.wisc.edu/~angenent/276/wallis.pdf

https://www.sciencefocus.com/science/archimedes-inventor-of-war-machines-and-calculus-almost/

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/

Modulo 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