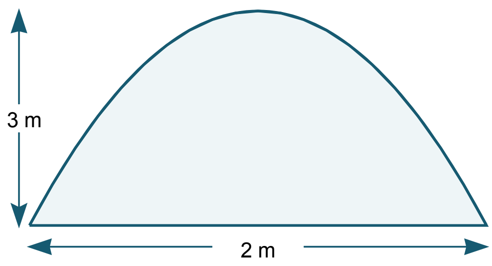
Universidade Federal do Espírito Santo

Departamento de Computação e Eletrônica - CEUNES
PROGRAMAÇÃO I / PROGRAMAÇÃO FUNCIONAL
Prof. Oberlan Romão

Exercício Programa 2

Calculo de área sobre curvas[1]

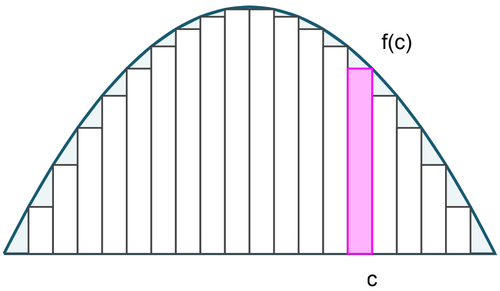
Sabemos, através da Geometria, como calcular áreas de polígonos. Esse conhecimento pode ser utilizado para o cálculo de áreas de regiões que possam ser divididas em um número finito de regiões. Entretanto, muitos casos não conseguimos decompor desse modo, de forma exata, a região que queremos calcular a área. Por exemplo, imagine uma parede construída no formato de arco parabólico, como mostrado na figura abaixo, e precisamos fornecer vidro para preenchê-la. Quantos de vidro será necessário?



Para responder a essa pergunta, precisamos conhecer a área sob a curva. Existem duas maneiras de fazer isso:

- Usar uma aproximação (encontrando áreas de retângulos);
- Usar integral definida;

A primeira alternativa é aproximar a área do arco dividindo o espaço em retângulos e somando a área desses retângulos, algo assim:

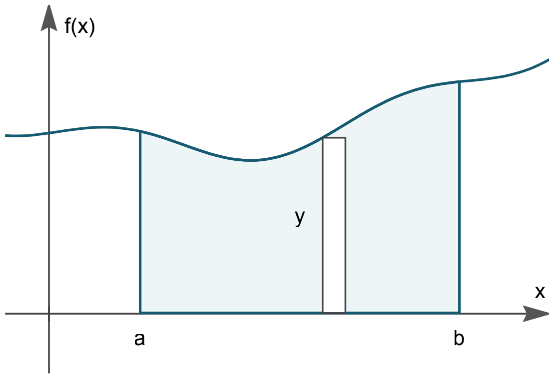


A altura de cada retângulo é encontrada calculando os valores da função, como mostrado para o caso típico $x = c$, onde a altura do retângulo é $f(c)$. Quanto maior o número de retângulos, melhor será a aproximação da área do arco. No diagrama acima, estamos aproximando a área usando retângulos

[1] Descrição baseada no site: https://www.intmath.com/integration/3-area-under-curve.php

internos (cada retângulo está dentro da curva). Também podemos encontrar a área usando os retângulos externos.

Outra maneira melhor é usar **integral definida**. Suponha que queiramos encontrar a área sob uma curva, como na figura abaixo (note que a curva está completamente acima do eixo x).



Quando Δx se torna extremamente pequeno, a soma das áreas dos retângulos fica cada vez mais próxima da área sob a curva. Se Δx tender a 0, obtemos a área exata. Podemos mostrar que, em geral, a área exata sob uma curva $y = f(x)$ de $x = a$ até $x = b$ é dado pela integral definida:

$$Area = \int_a^b f(x)dx$$

Como podemos avaliar essa expressão? Se $F(x)$ é a antidirivada de $f(x)$, então

$$\int_a^b f(x)dx = [F(x)]_a^b = F(b) - F(a)$$

Isso faz parte do *Teorema Fundamental do Cálculo.*

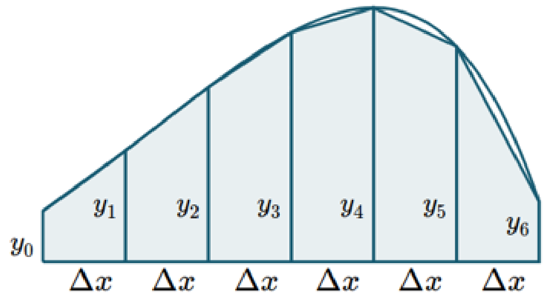
Exemplo: A equação da parábola do exemplo inicial é dada por $f(x) = -3x^2 + 6x$, com x dado em metros. Para calcularmos a área da parede, podemos usar integral definida:

$$\int_0^2 -3x^2 + 6xdx = \left[-x^3 + 3x^2\right]_0^2 = (-(2)^3 + 3(2)^2) - (0 + 0) = 4m^2$$

Como vimos, o procedimento usual de cálculo de uma integral definida é encontrar uma antiderivada do integrando e aplicar o Teorema Fundamental do Cálculo. Há situações, entretanto, em que é muito difícil ou impossível de encontrar a antiderivada. Nesses casos, é necessário obter uma **aproximação numérica da integral**. Dois métodos numéricos simples para se resolver uma integral definida é a *Regra dos Trapézios* e *Regra de Simpson.*

Regra dos Trapézios

Na Regra dos Trapézios, em vez de usar retângulos, como fizemos na nossa primeira tentativa para encontrar a área de um arco, usaremos trapézios e descobriremos que isso dá uma melhor aproximação à área.



A área aproximada sob a curva é encontrada somando a área de todos os trapézios. Portanto, a área total é dada por:

$$Area \approx \frac{1}{2}(y_0 + y_1)\Delta x + \frac{1}{2}(y_1 + y_2)\Delta x + \frac{1}{2}(y_2 + y_3)\Delta x + \dots$$

Dessa forma, a fórmula para o cálculo de integral pelo método do trapézio é dada por:

$$\int_a^b f(x)dx \approx \frac{\Delta x}{2}\left(f(x_0) + 2f(x_1) + 2f(x_2) + \dots 2f(x_{n-1}) + f(x_n)\right),$$

onde, $\Delta x = \frac{b-a}{n}$ e $x_i = a + i\Delta x$.

Uma informação importante, quando se está trabalhando com aproximações numéricas, é saber o quão distante/próximo o valor encontrado está do valor "exato". Para isso, podemos usar o erro relativo, que pode ser dado pela seguinte formula:

$$ErroRelativo = \frac{|ValorEncontrado - ValorIntegral|}{ValorIntegral},$$

onde, $ValorEncontrado$ é o valor encontrado pelo método e $ValorIntegral$ é o valor exato ou mais próximo possível (com maior número de casas decimais) do valor da integral definida.

Exemplo: Usando 10 subintervalos, calcular o valor da integral $\int_0^1 e^x dx$ usando a Regra dos Trapézios.
Solução: $\Delta x = \frac{1-0}{10} = 0.1$

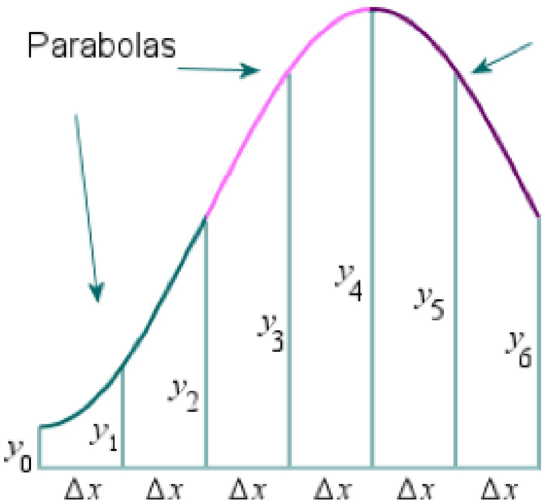
$$\int_0^1 e^x dx \cong \frac{0.1}{2}\left(e^0 + 2e^{0.1} + 2e^{0.2} + 2e^{0.3} + \dots + 2e^{0.7} + 2e^{0.8} + 2e^{0.9} + e^1\right) \cong 1.719713491389310$$

Sabendo que o valor da integral definida é dado por $\int_0^1 e^x dx = e^1 - e^0 \cong 1.718281828459050$, o erro relativo será:

$$ErroRelativo = \frac{|1.719713491389310 - 1.718281828459050|}{1.718281828459050} = 0.000833194477505.$$

Regra de Simpson

Na Regra de Simpson, usamos um polinômio interpolador de ordem 2 (parábola) para aproximar cada parte da curva. Em comparação com outros métodos numéricos que vimos, a regra de Simpson se mostra muito mais precisa.



Nós dividimos a área em n segmentos iguais de largura Δx. A área aproximada é dada pela seguinte formula:

$$Area = \int_a^b f(x)dx \approx \frac{\Delta x}{3}\left(f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \cdots + 4f(x_{n-1}) + f(x_n)\right),$$

onde, $\Delta x = \frac{b-a}{n}$ e $x_i = a + i\Delta x$.

Obs.: Na Regra de Simpson, n deve ser par.

Exemplo: Usando 10 subintervalos, calcular o valor da integral $\int_0^1 e^x dx$ usando a Regra de Simpson.
Solução: $\Delta x = \frac{1-0}{10} = 0.1$

$$\int_0^1 e^x dx \cong \frac{0.1}{3}\left(e^0 + 4e^{0.1} + 2e^{0.2} + 4e^{0.3} + \cdots + 4e^{0.7} + 2e^{0.8} + 4e^{0.9} + e^1\right) \cong 1.718282781924820$$

$$ErroRelativo = \frac{|1.718282781924820 - 1.718281828459050|}{1.718281828459050} = 0.00000055489487$$

Tarefa

Sua tarefa nesse EP é calcular o valor aproximado de integral definida para diferentes funções usando os dois métodos apresentados e fazer um relatório técnico com uma análise e discussão dos resultados obtidos. Após o programa ser feito, você deve preencher o arquivo Jupyter Nootebook (EP2.ipynb). Para cada aluno será atribuída, posteriormente, duas ou três funções e os valores de a e b para se calcular o valor da integral definida.

Para ficar mais interessante, você deve separar o seu programa em um módulo que contém, basicamente, as implementações dos métodos (integral.py) e um módulo principal (main.py).

Módulo integral

O módulo integral (integral.py) é responsável por calcular a integral definida:

$$\int_a^b f(x)dx,$$

usando a Regra dos Trapézios e Regra de Simpson. Esse módulo deve conter, no mínimo, três funções:

- Uma função para calcular o valor aproximado da integral definia usando a Regra dos Trapézios. Essa função deve ter os seguintes parâmetros obrigatórios: a função, o valor de a, o valor b e o valor de n (número de subintervalos). Se necessários, ela pode conter outros parâmetros. Você deve verificar se a, b e n são valores numéricos. A função deve retornar o valor aproximado da integral;

- Uma função para calcular o valor aproximado da integral definia usando a Regra de Simpson. Essa função deve ter os seguintes parâmetros obrigatórios: a função, o valor de a, o valor b e o valor de n (número de subintervalos). Se necessários, ela pode conter outros parâmetros. Você deve verificar se a, b e n são valores numéricos. A função deve retornar o valor aproximado da integral;

- Uma função para calcular o valor aproximado da integral para diferentes valores de n. Essa função deve ter os seguintes parâmetros obrigatórios: o método a ser usado, a função, o valor de a, o valor b e uma lista com diferentes valores de n. A função deve retornar uma lista contendo o valor aproximado da integral para cada valor de n.

Dessa forma, esse módulo será composto por algo assim (não use esses nome para as funções):

```
def m1(f, a, b, n):
    #implementação da Regra dos Trapézios

def m2(f, a, b, n):
    #implementação da Regra de Simpson

def abc(funcaoMetodo, f, a, b, L):
    #implementação da terceira função
```

Perceba que essas três funções são de alta ordem.

Módulo main

O módulo main (main.py) é responsável por chamar o módulo anterior e apresentar as saídas necessárias para o relatório.

Esse módulo deve conter, no mínimo, as seguintes funções:

- Uma função associada a cada função que se deseja calcular o valor da integral definida;

- Uma função `main` que, basicamente, exibe o valor da integral definida e o erro relativo de cada função para cada método;

Dessa forma, esse módulo será por algo assim:

```
import math
...
def funcao(x):
  return math.exp(x)

def main():
  ...
  #Calcula o valor da integral e^x com a = 0 e b = 1 para diferentes valores de n;
  a = 0; b = 1;
  N = [10, 20, 30]
  L = abc(m1, funcao, a, b, N) #Usando a Regra dos Trapézios
  #imprime os valores de L e o erro
  L = abc(m2, funcao, a, b, N) #Usando a Regra de Simpson
  #imprime os valores de L e o erro
```

```
    ...

main()
```

Sempre que for imprimir valores do tipo ponto flutuante, imprima com, pelo menos, 20 casas decimais.

O que entregar

Você deve entregar, **via AVA**, os códigos-fonte (arquivo integral.py e main.py), um relatório (formato pdf) e o arquivo Jupyter Nootebook (EP2.ipynb) preenchido.

Data de entrega: até às 6h do dia 09/07/2019.

Observações:

1. Não é permitido usar **estruturas de repetição (loop)**, como `while`, `for` e **funções impuras**. A utilização dessas estruturas/funções implicará em nota 0. Quando necessário, utilize funções recursivas;
2. Não use variáveis globais para evitar a possibilidade de uma função se tornar impura;
3. Use apenas instruções/comandos visto em sala de aula (teórica ou prática). Se quiser usar algo que ainda não foi visto, me pergunte;
4. Documente o seu programa: comente e use `"""docstring"""` nas funções. Código sem comentários e/ou sem `"""docstring"""` valerá, no máximo, 9,0 pontos.
5. **Em caso de plágio, será atribuído 0 a todos os envolvidos.**

Critérios de Avaliação

A nota do EP se dará pela seguinte fórmula:

$$R \times S \times NB \times (N_{EP}),$$

onde,

- $R = \begin{cases} 1, & \text{se enviou o relatório técnico;} \\ 0, & \text{caso contrário (ou se enviou um arquivo em branco).} \end{cases}$
- $S = \begin{cases} 1, & \text{se enviou os } \textit{scripts} \text{ com as implementações solicitadas;} \\ 0, & \text{caso contrário (ou se enviou um arquivo em branco).} \end{cases}$
- $NB = \begin{cases} 1, & \text{se enviou o Jupyter Notebook;} \\ 0, & \text{caso contrário (ou se enviou um arquivo inalterado).} \end{cases}$
- N_{EP}: Nota geral do EP, sendo $0.0 \leq N_{EP} \leq 10.0$, detalhado abaixo.

A nota geral do EP (N_{EP}) será avaliada da seguinte forma:

- Relatório técnico: 4.0 pontos (caprichem). O relatório deve conter, pelo menos, os seguintes pontos:
  - Apresentação das funções que serão calculado o valor da integral definida juntamente com o gráfico da função (usando o `Matplotlib`). Para cada aluno será atribuída duas ou três funções e os valores de a e b;

- Tabelas com os valores gerados pelo seu programa;
- Análise e discussão dos resultados obtidos pelos dois métodos, mostrando vantagens e desvantagens de cada método. Você pode comparar, por exemplo, qual dos métodos se aproxima mais rápido para o valor da integral, qual método é mais simples e/ou eficiente, etc.;
- Gráficos (feitos com o `Matplotlib`) comparando os resultados.

• Jupyter Nootebook: 1.0 ponto;

• Implementação dos métodos: 5.0 pontos;

  - Método dos Trapézios: 1.5 pontos;
  - Método de Simpson: 1.5 pontos;
  - Terceira função do módulo integral definida no EP: 1.5
  - Função `main()`: 0.5 ponto;

Perceba que se não for enviado um dos arquivos (relatório técnico ou arquivo .py ou o Jupyter Notebook), sua nota será 0.

Exemplo de saída

```
Função: e^x com a = 0 e b = 1
Valor Exato = 1.71828182845905685916

==> Regra dos Trapázios
---------------------------------------------------------------
n                integral                        erro
---------------------------------------------------------------
  4        1.72722190455751656302        0.00520291604694271895
 10        1.71971349138931461908        0.00083319447749829567
 50        1.71833910413815749152        0.00003333311110669010
100        1.71829614745041747703        0.00000833331943774267
150        1.71828819246110287011        0.00000370370095324708
200        1.71828540821136321881        0.00000208333245866305
300        1.71828341946044371724        0.00000092592574774819
400        1.71828272339740384389        0.00000052083327202925
500        1.71828240121961672848        0.00000033333330445738
600        1.71828222620944925936        0.00000023148146352505
700        1.71828212068383767708        0.00000017006801560602
800        1.71828205219365215406        0.00000013020832298247
900        1.71828200523700580504        0.00000010288064857464
---------------------------------------------------------------
==> Regra de Simpson
---------------------------------------------------------------
n                integral                        erro
---------------------------------------------------------------
  4        1.71831884192174699777        0.00002154097312623740
 10        1.71828278192482342135        0.00000055489486693650
 50        1.71828182998633427481        0.00000000088883988083
100        1.71828182855450384281        0.00000000005554792123
150        1.71828182847790200682        0.00000000001096743698
200        1.71828182846501187342        0.00000000000346567959
300        1.71828182846022370356        0.00000000000067907626
400        1.71828182845941723755        0.00000000000020973183
500        1.71828182845919763544        0.00000000000008192852
600        1.71828182845911836552        0.00000000000003579527
700        1.71828182845908683518        0.00000000000001744535
800        1.71828182845906951570        0.00000000000000736581
900        1.71828182845906152210        0.00000000000000271372
---------------------------------------------------------------
```

Bom trabalho!!!