Lab 4 – Interpolação e Integração (Volume)

Material necessário:

- Recipiente de volume conhecido;

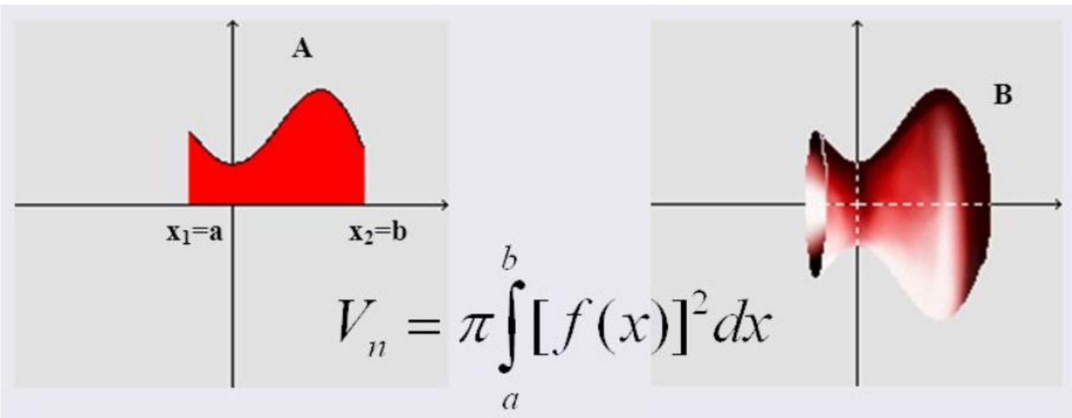
-Objeto de dimensões conhecidas;

-Celular com câmera

Experiência:

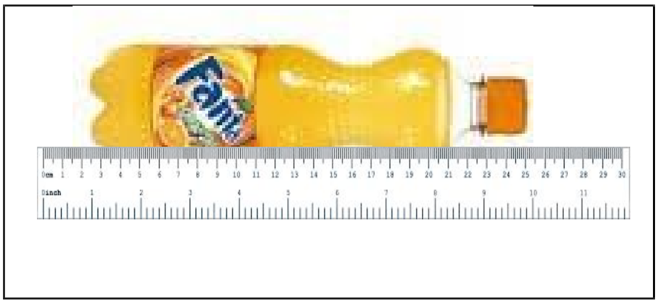
O objetivo é encontrar o volume de um recipiente com formato aproximadamente cilíndrico utilizando técnicas de interpolação e de integração numéricas. Para tal, iremos aproximar o perfil do recipiente por polinômios interpoladores f(x). Em seguida podemos calcular o volume do sólido de revolução através da seguinte integral:

$$V = \pi * \int f(x)^2 \, dx$$



Fonte: https://slideplayer.com.br/slide/49608/

O recipiente utilizado deve possuir, de preferência, oscilações em seu perfil (garrafa de fanta, por exemplo). O grupo deve fotografar o recipiente de lado, de forma que o eixo mais comprido desde esteja alinhado com o eixo mais comprido da foto (aconselhamos apoiar o celular em uma mesa para tirar a foto), como mostrado na figura abaixo. Para poder calcular o volume do recipiente em litros, se faz necessário ter uma relação pixel/cm da foto. Para tal, inclua na imagem, o mais próximo possível do recipiente, algum objeto de dimensões conhecidas, como uma régua, por exemplo.



Após adquirir a foto, o grupo deve utilizar o paint para marcar, no mínimo, 9 subintervalos (10 pontos) na parte superior do contorno. **Tais pontos deve ser igualmente espaçados.** Os pontos devem ser tais que o eixo central da garrafa seja considerado a origem do eixo y (ou seja, o componente y dos pontos marcados devem ser a distância do contorno ao centro do recipiente). O origem do eixo x é irrelevante. Também colete os pontos que delimitam o objeto de dimensões conhecidas.

	Ponto 1	Ponto 2	Ponto 3	Ponto 4	Ponto 5	Ponto 6...
X (px)						
Y (px)						

A partir da tabela de pontos coletados, faça o que se pede:

1- Calcule a relação pixel/cm utilizando o objeto de dimensões conhecidas.
2- Utilize os método de integração do trapézio e segunda regra de simpson para estimar o volume do recipiente, em litros, utilizando os pontos coletados. Perceba que neste caso as funções de integração recebem apenas a tabela de pontos x e y (dois vetores).
3- Utilize a interpolação de Lagrange para calcular a função interpoladora para aproximar o contorno da garrafa. Plote tal função sobre a imagem obtida. Comente a qualidade da aproximação.
4- Utilize a função interpoladora de Lagrange obtida para achar o volume do recipiente, utilizando o método da primeira regra de simpson, com um número elevado de subintervalos. Neste caso, a função de integração irá integrar uma função.
5- Utilize a interpolação de Newton para calcular vários polinômios interpoladores (interpolação por partes). Agrupe de dois em dois **subintervalos**, observando que cada grupo produzirá um polinômio de grau 2. (Se o número total de subintervalos for ímpar, reduza o grau do último polinômio). Plote os polinômios obtidos nas imagens. Comente a qualidade da aproximação.

6- Utilize o conjunto de funções interpoladoras de newton para calcular o volume do recipiente utilizando os métodos do trapézio, primeira e segunda regras de simpson, mas agora com um número elevado de subintervalos. Neste caso as funções de integração irão integrar funções.
7- Compare e comente todos os resultados encontrados entre si e com o volume real do recipiente.

Programa:

O algoritmo abaixo implementa o método da regra dos trapézios composta para encontrar a integral de uma função f(x) já conhecida. Observe que em alguns casos não conhecemos f(x), mas diretamente os pares de pontos (x,y), o que pressupõe alteração do código.

Algoritmo Trapézio Composto

Entrada: Intervalo da integral [a,b], número de subintervalos n (a função que iremos achar a integral será declarada em f).
Saída: Valor da integral S.

```
h=(b-a)/n
x(1)=a
y(1)=f(x(1))     // f é função cuja integral será calculada
S=y(1)
para  i=2 até n
        x(i)=x(i-1)+h
        y(i)=f(x(i))
        S=S+2*y(i)
fim_para
x(n+1)=b
y(n+1)=f(x(n+1))
S=h/2*(S+y(n+1))
```

Teste do código:
a=0;
b=%pi/2;
f(x)= sin(x); //definir essa função no sistema

Chamada: I=trapC(a,b,1)
Saída: S= 0.7853982;

Chamada: I=trapC(a,b,10)
Saída: S= 0.9979430;

Chamada: I=trapC(a,b,100)
Saída: S= 0.9999794;

Algoritmo de Lagrange

Entrada: Vetores coluna "x" e "y" dos pares de pontos da função a ser interpolada e ponto "p" onde a função interpoladora será avaliada.
Saída: valor S da função interpoladora no ponto "p".

```
n = comprimento(x) //no Scilab n=length(x)
S = 0;
para i = 1 até n
    L = 1
    para j = 1 até n
        se (i~=j)
                L = L*(p-x(j))/(x(i)-x(j))
        fim_se
    fim_para
    S = S+L*y(i);
fim_para
```

Teste do código:

```
x=[1 7 14 16 23 25];
y=[ 19  - 4  - 33  - 40  - 71  - 75]

Chamada:  S=lagrange(x,y,20)
Saída: S= - 57.382696;

cont=1;
for i=x(1):0.1:x(length(x))
s(cont)=lagrange(x,y,i);
cont=cont+1;
end
plot(x,y,'.r')
plot(x(1):0.1:x(length(x)),s','b')
```

Algoritmo das diferenças divididas

Entrada: Vetores coluna x e y dos pares de pontos
Saída: Tabela Tab com as diferenças divididas (de ordem zero até a ordem máxima)

```
n = comprimento(x)
Tab(:,1)=y;   //diferenças divididas de ordem zero
para i = 1 ate n-1
        para j = 1 ate n-i
        Tab(j,i+1) = (Tab(j+1,i)-Tab(j,i))/(x(j+i)-x(j));
        fim_para
fim_para
```

```
x=[ 1 5 7 12 25];
y=[ 3 -5 7 -9 11];
Chamada:  t=diffdiv(x,y)
Saída:
t  = [3,   - 2,    1.333,  - 0.240,        0.0133;
  - 5,     6,   - 1.314,    0.0789,        0;
    7,  - 3.2,        0.263,    0,         0;
  - 9,     1.538,     0,        0,         0;
    11,   0,          0,        0,         0]
```

Algoritmo da interpolação de Newton

Entrada: Vetores coluna "x" e "y" dos pares de pontos da função a ser interpolada e ponto "p" onde a função interpoladora será avaliada.
Saída: Valor S da função interpoladora no ponto "p"

```
n = comprimento(x)
Tabdiffdiv = diffdiv(x,y); //funcao diffdiv definida acima
S = y(1);
para i=2 ate n
        M = 1
        para j = 1 ate i-1
        M = M*(p - x(j))
        fim_para
        S = S + M*Tabdiffdiv(1,i)
fim_para
```

```
x=[ 1 5 7 12 25];
y=[ 3 -5 7 -9 11];

Chamada:  s=newton(x,y,20)
Saída:  s= - 152.09812


cont=1;
for i=x(1):0.1:x(length(x))
    s(cont)=newton(x,y,i);
    cont=cont+1;
end

plot(x,y,'.r',x(1):0.1:x(length(x)),s', 'b')
```