

Unidade 4: Utilização da biblioteca TI PlotLib

Aplicação: Posições sucessivas num movimento

Nesta aplicação da Unidade 4 vamos aprender como estudar um movimento a partir das medições efetuadas por um cronofotógrafo e visitar o conhecimento adquirido, nas três lições anteriores, na utilização da biblioteca **TI PlotLib**.

Objetivos:

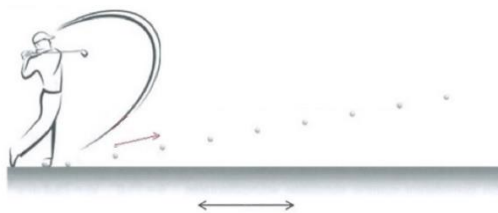
- Representar uma nuvem de pontos.
- Efetuar o cálculo, seguido da representação dos vetores de um sistema modelado por um ponto.

PROBLEMA:

Estudemos o movimento de uma bola de golfe a partir de uma cronofotografia ou fotografia estroboscópica (registo do movimento de um objeto tirando fotos num intervalo de tempo fixo, observando a sua posição).

Pretende-se representar a evolução do vetor velocidade ao longo do tempo.

As posições são lidas a cada 0,066 segundos e registadas numa tabela.



t (s)	0	0.066	0.132	0.198	0.264	0.33	0.396	0.462	0.528	0.594	0.66
x (m)	0.01	0.25	0.57	0.91	1.22	1.54	1.87	2.16	2.49	2.81	3.15
y (m)	0.015	0.34	0.681	1.01	1.297	1.559	1.768	1.95	2.08	2.158	2.193

NOTA:

Na Unidade 5, deste projeto “10 Minutos de Código”, poderá aprender a importar dados de listas da calculadora utilizando o módulo **TI System**.

IMPLEMENTAÇÃO:

Etapla 1: Introdução das medidas registadas e criação da lista com os valores dos tempos de registo.

- Adicionar um novo programa TI-Python com o nome U4AP.
- Importar a biblioteca **TI PlotLib** para se utilizar as instruções de representação gráfica.
- Definir a variação do tempo de registo de dados, **dt=0.066**.
- Inserir as medidas correspondentes às coordenadas, **x** e **y**, dos pontos identificados durante a análise da cronofotografia.

```
U4AP_PT
U4AP.py guardado com sucesso
import ti_plotlib as plt
# Dados do problema
dt=0.066
x=[0.01,0.25,0.57,0.91,1.22,1.54,1.87,2.16,2.49,2.81,3.15]
y=[0.015,0.34,0.681,1.01,1.297,1.559,1.768,1.95,2.08,2.158,2.193]
# Vetores velocidade
vx=[]
vy=[]
```

Etapla 2: Cálculo dos vetores velocidade

- O vetor velocidade, num dado instante t_i , é dado pela igualdade:
- $$\vec{V}_i = \frac{M_i - M_{i-1}}{t_i - t_{i-1}}, \text{ sendo } M_i \text{ a posição da bola no instante } t_i.$$
- Desta forma, portanto, não será representado o vetor velocidade do último ponto da lista, facto que dever-se-á ter em atenção no código.

```
*U4AP_PT
*U4AP.py 13/13
dt=0.066
x=[0.01,0.25,0.57,0.91,1.22,1.54,1.87,2.16,2.49,2.81,3.15]
y=[0.015,0.34,0.681,1.01,1.297,1.559,1.768,1.95,2.08,2.158,2.193]
# Vetores velocidade
vx=[]
vy=[]
n=len(x)
for i in range(0,n-1):
    vx.append((x[i+1]-x[i])/dt)
    vy.append((y[i+1]-y[i])/dt)
escala=2
```

DICA:

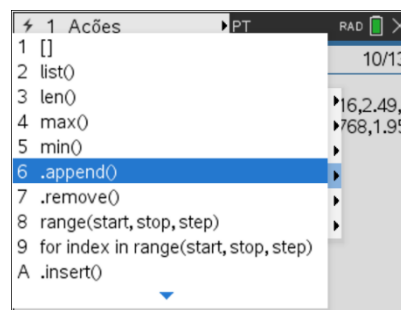
Para tornar a representação gráfica legível, usaremos 2 como fator de escala.





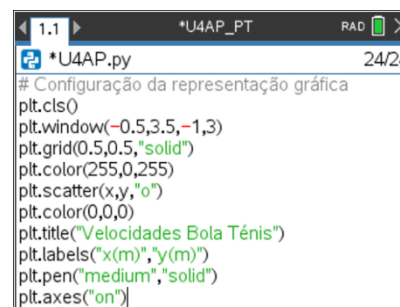
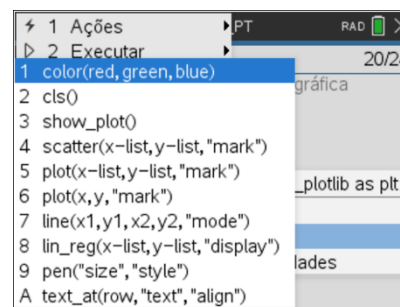
RECORDE QUE:

- A função **len(x)** permite obter o número de elementos da lista x, isto é, a sua dimensão (comprimento). Poderá escrever diretamente no programa o nome da função ou aceder ao menu **4: Planos integrados**, seguido de **4: Listas** e por fim **3: len()**.
- A instrução **.append()** permite acrescentar, no fim da lista, um elemento a uma dada lista e que se pode obter no menu **4: Planos integrados**, seguido de **4: Listas** e por fim **6: append()**.



Etapa 3: Definição dos parâmetros da representação gráfica

- **plt.cls()** – para limpar o ecrã.
- **plt.title("title")** – para inserir um título no gráfico ("Velocidades Bola Ténis").
- **plt.window(xmin, xmax, ymin, ymax)** – para definir a janela de visualização da representação gráfica $([-0.5, 3.5] \times [-1, 3])$.
- **plt.grid(x-scale, y-scale, "type")** – para mostrar uma grelha de fundo com linha contínua e graduação 0.5 $((0.5, 0.5, "solid"))$.
- **plt.color(255,0,255)** – para definir a cor magenta para a nuvem de pontos.
- **plt.scatter(x-list, y-list, "mark")** – para definir as listas e a marca da nuvem de pontos $(x, y, "o")$.
- **plt.color(0,0,0)** – para definir como preto da cor dos eixos coordenados.
- **plt.pen("size", "style")** – para definir os eixos com espessura média e linha contínua $("medium", "solid")$.
- **plt.labels("x-label", "y-label")** – para definir as etiquetas dos eixos coordenados, localizadas por defeito nas linhas 12 e 2, respetivamente $(x(m), y(m))$.

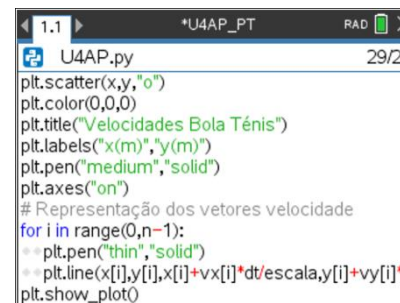


A representação gráfica dos vetores velocidade é efetuada através de um ciclo limitado de n-1 valores, com as instruções:

- **plt.pen("size", "style")** – para traçar os vetores com espessura estreita e linha contínua $("thin", "solid")$.
- **plt.line(x0,y0,x1,y1)** – para traçar um vetor velocidade.

No fim do ciclo, é necessário tornar visível todos os elementos gráficos através da função:

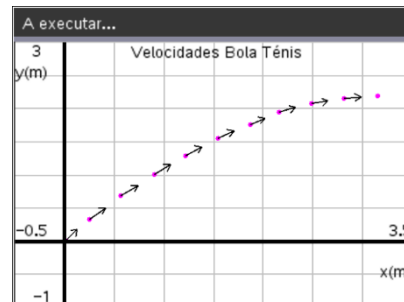
- **plt.show()** – para mostrar toda a representação gráfica.



RECORDE QUE:

A posição da bola de ténis num instante t , entre os instantes t_i e t_{i+1} , pode ser identificada num referencial cartesiano por $x_i + v_{x_i} \times dt$ para a abcissa e $y_i + v_{y_i} \times dt$ para a ordenada, com $dt = t_{i+1} - t_i$. Note que, se o instante t for igual a t_{i+1} , então a posição da bola tem as coordenadas (x_{i+1}, y_{i+1}) , já que $dt = t_{i+1} - t_i = 0.066$.

- Execute o programa, clicando simultaneamente nas teclas **Ctrl** e **R**.
- Deverá obter uma representação análoga à do ecrã ao lado. Observe a representação e associe cada linha de código aos elementos representados.



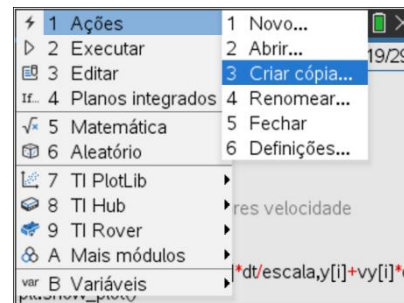
NOTE QUE:

O facto escala utilizado no programa, no caso de 2, permite representar um vetor colinear com o vetor velocidade e tornar a representação mais perceptível.

OBSERVAÇÕES:

- Para cortar, copiar ou colar uma linha ou uma instrução, use os atalhos **Ctrl + X**, **Ctrl + C** e **Ctrl + V**, respetivamente.

- Para programas mais complexos e/ou mais extensos, poderá optar por duplicar o programa como estratégia de reutilização de trabalho realizado. Para isso, com o editor de programas do TI-Python aberto no programa a duplicar, pressione a tecla **menu**, em seguida na opção **1: Ações** e por fim seleciona a opção **3: Criar cópia ...**. Surgirá uma janela para inserir o nome.



- Se a quantidade de dados a utilizar for grande ou proveniente de uma atividade experimental realizada com uma consola de recolha de dados, é possível usar a aplicação "Listas e Folha de Cálculo" da TI-Nspire CX II para copiar dados facilmente e guardá-los em forma de listas antes de importá-los (consulte a Lição 1 da Unidade 5).