Lektion 4 : Verwendung des Moduls ti_plotlib

Anwendung : Bewegungsablauf

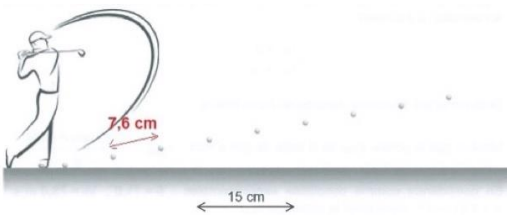
In der Anwendung der Lektion 4 geht es darum, Daten aus der Photografie einer Bewegung zu gewinnen und unter Verwendung des Moduls **ti_plotlib** grafisch darzustellen..

Lernziele :

- Darstellung einer Punktwolke.
- Berechnung und Darstellung von Bewegungsvektoren eines durch Punkte modellierten Systems.

Die Aufgabe : Die Bewegung eines Golfballes soll untersucht werden, der in regelmäßigen Abständen auf einem Photo abgebildet wurde. Dabei soll es um den zeitlichen Verlauf des Geschwindigkeitsvektors gehen. Die einzlnen Aufnahmen haben untereinander einen zeitlichen Abstand von 0,066 s und sind unten in einer Tabelle dargestellt.

Anmerkung: In Lektion 5 wird gezeigt, wie man mit dem Modul **ti_system** Messungen aus Rechnerlisten importieren kann.

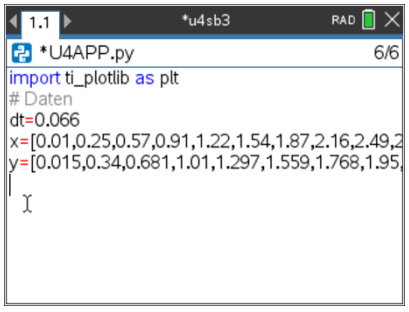


t(s)	0	0.066	0.132	0.198	0.264	0.33	0.396	0.462	0.528	0.594	0.66
x(m)	0.01	0.25	0.57	0.91	1.22	1.54	1.87	2.16	2.49	2.81	3.15
y(m)	0.015	0.34	0.681	1.01	1.297	1.559	1.768	1.95	2.08	2.158	2.193

Das Programm :

1 : Eingabe der Messwerte und der Zeit.

- Das neue Programm erhält den Namen **U4APPS**.
- Das Modul **ti_plotlib** wird importiert.
- Eingabe der Messwerte durch Auswertung des Photos.

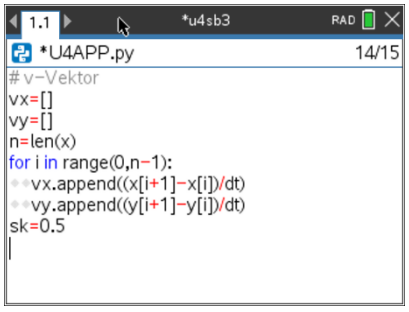


2 : Bestimmung des Geschwindigkeitsvektors :

Der Geschwindigkeitsvektor v_i im Intervall $[t_i ; t_{i+1}]$ berechnet sich durch :

$$\vec{V_i} = \frac{\overrightarrow{M_i M_{i+1}}}{t_{i+1} - t_i}$$

Für den letzten Messpunkt lässt sich der Vektor offenbar nicht berechnen.

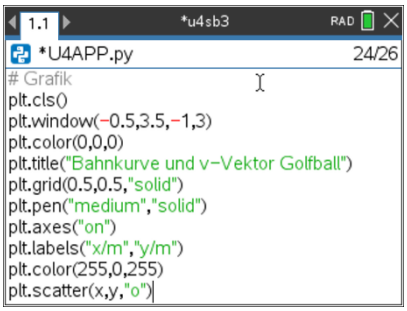


Hinweis 1 : Zur besseren Lesbarkeit der grafischen Darstellung wird ein Skalierungsfaktor sk = 0.5 für die Länge der Geschwindigkeitsvektoren verwendet.

Hinweis 2 : Der Befehl **len(x)** liefert die Anzahl der Elemente der Liste **x**.

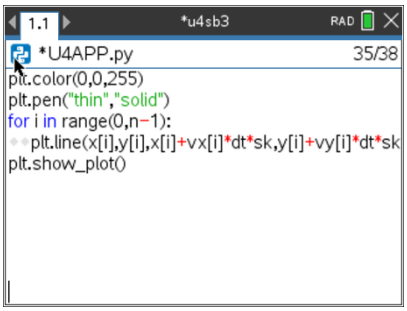
3 : Einstellungen der Grafik :

- **plt.cls()** löscht den Bildschirm.
- **plt.title(« title »)** gibt der Grafik einen Namen.
- **plt.window(x_{min}, x_{max}, y_{min}, y_{max})** stellt das Grafikfenster ein.
- **plt.grid(xscl, ysc, « type »)** erzeugt ein Gitternetz mit dem Abstand 0,5.
- **plt.color(255,0,255)** stellt die Farbe Magenta dar.
- **plt.scatter(x, y, « o »)** stellt die Punktwolke der Messwerte dar.
- **plt.color(0,0,0)** lässt die Achsen in schwarz zeichnen.
- **plt.pen(« medium », « solid »)** stellt die Zeicheneigenschaften ein.
- **plt.labels(« x(m) », « y(m) »)** beschriftet die Achsen an den Standardstellen 12 rechts und 2 links.

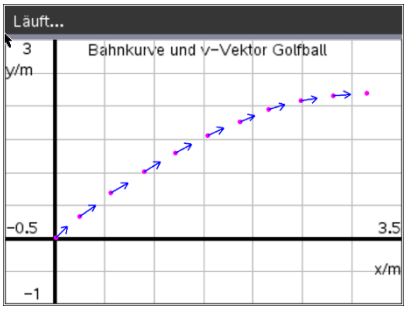


4 : Die Vektoren werden durch eine For-Schleife mit n-1 Elementen dargestellt :

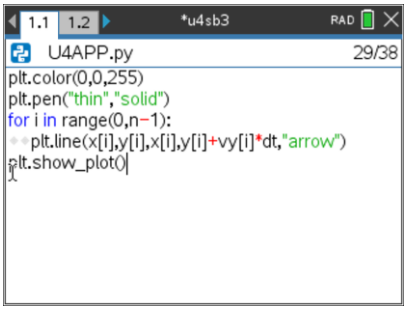
- **plt.line(x0,y0,x1,y1, »arrow »)** Länge und Endstil des Geschwindigkeitsvektors.
- **plt.show()** zur dauerhaften Darstellung der Grafik.



Lässt man das Programm laufen, sollte sich eine Grafik wie nebenstehend ergeben.



Das Bild ganz unten zeigt nur die y-Komponente des v-Vektors (bei sk=1 ; man beachte die nebenstehenden Änderungen in der Anweisung plt.line !). Man sieht sehr schön, dass sie zum Scheitelpunkt der Bahnkurve abnimmt.



Lehrertipp: Unter **Bearbeiten** steht eine Toolbox mit Kopier- und Einfügeanweisungen zur Verfügung, um die Bearbeitung eines Programmes zu erleichtern.
Bei komplexen Programmen hat man auch die Möglichkeit, ein Programm zu duplizieren. Man findet diese Möglichkeit unter **Aktionen**. Dort muss dann lediglich ein neuer Name für das duplizierte Programm vergeben werden.