La traiettoria del proiettile

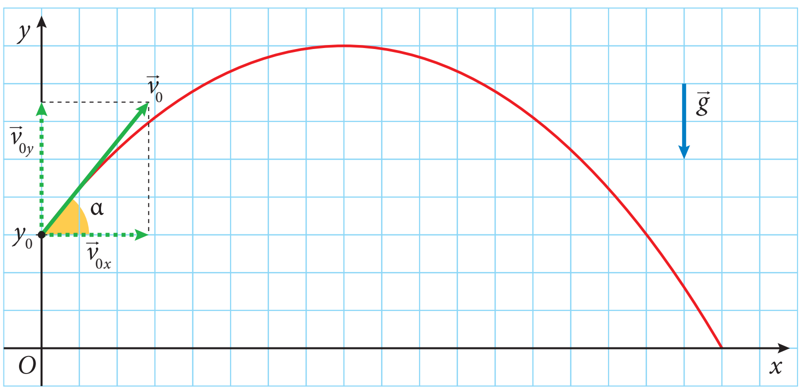
Scarica il file .py

OBIETTIVO Scrivere un programma per simulare il moto di un proiettile e tracciare il grafico della sua traiettoria a partire dalle condizioni iniziali del moto.

LA FISICA COINVOLTA

Il moto del proiettile soggetto all'accelerazione di gravità è la composizione di due moti in direzioni perpendicolari tra loro: un moto rettilineo uniformemente accelerato lungo la direzione perpendicolare al suolo e un moto rettilineo uniforme lungo la direzione parallela al suolo.

Utilizziamo il sistema di riferimento mostrato in figura, nel quale l'asse x è parallelo e radente al suolo.



Il proiettile viene lanciato a velocità v_0 lungo una direzione che forma un angolo α con l'asse x. Il punto in cui il proiettile è lanciato ha una altezza dal suolo pari a y_0. Senza perdere di generalità, supponiamo che l'ascissa del punto in cui è lanciato il proiettile sia nulla e che il proiettile sia lanciato all'istante $t = 0$ s

Nel sistema di riferimento definito sopra, se indichiamo con x e y le coordinate del proiettile al tempo t, abbiamo:

$$x = v_{0x}t \qquad [P1]$$

$$y = y_0 + v_{0y}t - \frac{1}{2}gt^2 \qquad [P2]$$

nella quali v_{0x} e v_{0y} sono le componenti cartesiane del vettore velocità iniziale $\vec{v}_0$. Queste componenti possono essere espresse in funzione dell'angolo di lancio α e del modulo della velocità iniziale v_0:

$$v_{0x} = v_0 \cos \alpha \qquad [P3]$$

$$v_{0y} = v_0 \sin \alpha \qquad [P4]$$

Ora abbiamo a disposizione tutti gli elementi di fisica per poter creare una simulazione del fenomeno in Python.

IL PROGRAMMA IN PYTHON PASSO DOPO PASSO

Iniziamo a scrivere il codice (file traiettoria_proiettile.py) inserendo le righe che importano i moduli Matplotlib (riga 1) e Numpy (riga 2).

```
1 import matplotlib.pyplot as plt
2 import numpy as np
```

Inizializziamo tutte le variabili e gli array necessari alla simulazione.

```
4  #==============================================================================
5  #DATI INIZIALI
6  g = 9.81 #m/s^2
7  alfa = 30.00*np.pi/180 #rad
8
9  y0 = 1.00 #m
10 v0 = 20.00 #m/s
11
12 v0_x = v0*np.cos(alfa) #m/s
13 v0_y = v0*np.sin(alfa) #m/s
14
15 durata = 2.14 #s
16 numero_istanti = 500 #numero di istanti in cui è suddiviso l'asse del tempo
17 t = np.linspace(0,durata,numero_istanti) #s
18 #==============================================================================
```

Nel codice, riportato sopra, definiamo le variabili seguenti.

- Alla riga 6 assegniamo a g il valore dell'accelerazione di gravità, in questo caso quella terrestre.
- Alla riga 7 definiamo `alfa` che contiene il valore dell'angolo di lancio α, in questo caso 30°. Poiché vogliamo esprimere l'angolo in radianti, dopo il simbolo di assegnamento = troviamo la formula di conversione da gradi a radianti.
- La riga 9 assegna a `y0` il valore dell'altezza y_0 in metri dalla quale è lanciato il proiettile.
- La riga 10 assegna alla variabile `v0` il valore della velocità iniziale v_0 del proiettile in m/s.

Queste variabili definiscono completamente la fisica del problema, ma per implementare la simulazione ci occorrono anche:

- le componenti v_{0x} e v_{0y} del vettore velocità iniziale, calcolate dalle righe 12 e 13 che traducono in Python le equazioni `[P3]` e `[P4]`;
- la durata di osservazione del moto, il cui valore in secondi è salvato nella variabile durata di riga 15;
- il numero di istanti `numero_istanti` in cui suddividiamo l'intervallo temporale di osservazione (riga 16). La simulazione calcolerà la posizione del proiettile in ognuno di questi istanti;
- l'array `t`, definito a riga 17, che contiene gli istanti di tempo di cui sopra: esso è generato prendendo un numero pari a `numero_istanti` di valori equi-spaziati tra `0` e `durata`.

Procediamo con l'implementazione delle equazioni del moto `[P1]` e `[P2]` nelle righe 22 e 23.

```
20 #==============================================================================
21 #EQUAZIONI DEL MOTO
22 x = v0_x*t
23 y = y0 + v0_y*t - 1/2*g*t**2
24 #==============================================================================
```

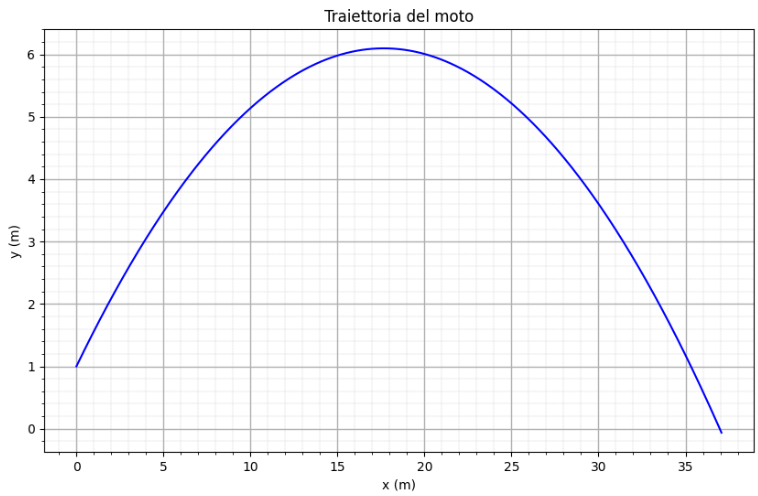
Queste due righe di codice restituiscono gli array x e y che contengono le coordinate del proiettile per tutti gli istanti di tempo salvati nell'array `t`. A partire da essi generiamo il grafico della traiettoria del moto usando il codice seguente.

```
26 #==============================================================================
27 #CREAZIONE DEL GRAFICO
28 fig, grafico = plt.subplots(figsize=(10,6))
29 grafico.plot(x, y, color='blue')
30 grafico.set_xlabel("x (m)")
31 grafico.set_ylabel("y (m)")
32 grafico.set_title("Traiettoria del moto")
33 grafico.minorticks_on()
34 grafico.grid(which='major', axis='both', linewidth=1.0)
35 grafico.grid(which='minor', axis='both', linestyle ="--", linewidth=0.2)
36 plt.show()
37 #==============================================================================
```

Dopo aver inizializzato gli oggetti `figura` e `grafico` nella riga 28, la riga 29 crea il grafico della traiettoria mediante la funzione `grafico.plot` che rappresenta i valori dell'array `y` rispetto a quelli dell'array `x`: la curva ottenuta è la traiettoria del proiettile. Le righe da 30 a 35 aggiungono etichette per gli assi, titolo e griglia. Infine, la riga 35 visualizza il grafico all'interno della finestra di output.

I RISULTATI

Quando eseguiamo il programma, la finestra di output contiene il grafico seguente.



La curva blu rappresenta la traiettoria del proiettile, dall'istante di lancio ($t = 0$ s) in cui il moto parte dal punto di coordinate $x = 0$ m e $y = 1$ m, per tutta la durata di osservazione del fenomeno, il cui valore è assegnato alla variabile `durata` inizializzata a riga 15. Osserviamo che la curva è un arco di parabola.

Muovendo il cursore sul grafico, possiamo posizionarlo sui punti di interesse della traiettoria e leggere le loro coordinate nella parte in basso a destra della finestra di output. Per esempio, individuiamo lo spostamento del proiettile lungo l'asse orizzontale leggendo la coordinata x del punto in cui la traiettoria tocca il suolo posto a $y = 0$ m: in questo caso si ottiene circa 36,97 m.

SPERIMENTIAMO

Possiamo sperimentare agendo sulle condizioni iniziali del moto del proiettile, ovvero variando la sua velocità iniziale (riga 10), l'angolo di lancio (riga 7) e l'altezza del punto da cui è effettuato il lancio (riga 9). Di volta in volta dobbiamo poi individuare la durata ottimale per osservare la parte di traiettoria di interesse: eseguendo alcune volte il programma possiamo trovare empiricamente il valore più opportuno da assegnare alla variabile `durata` di riga 15.

ESERCIZIO

Un esperimento di lancio del proiettile viene eseguito su tutti i pianeti rocciosi del sistema solare. Per tutte le esecuzioni dell'esperimento si ha che $v_0 = 10$ m/s, $y_0 = 0$ m e $\alpha = 50°$.

- Usando il programma, calcola la gittata G sui diversi pianeti.
- Verifica che il prodotto $g \cdot G$, dove g è l'accelerazione di gravità sulla superficie del pianeta, è costante e non dipende dal pianeta sul quale si esegue l'esperimento.

Possiamo migliorare il programma facendo in modo che visualizzi automaticamente solo la parte di traiettoria compresa tra il lancio e il momento in cui il proiettile colpisce il suolo, ovvero soltanto per il tempo di volo del proiettile.

Le variazioni da apportare sono descritte di seguito (file traiettoria_proiettile_tvolo.py). Per prima cosa fissiamo un valore di durata sufficientemente lungo da essere sicuri che sia maggiore del tempo di volo (riga 15 del codice sotto) e poi aumentiamo il numero di istanti di tempo (riga 16).

```
15 durata = 100.00 #s
16 numero_istanti = 10000 #numero di istanti in cui è suddiviso l'asse del tempo
```

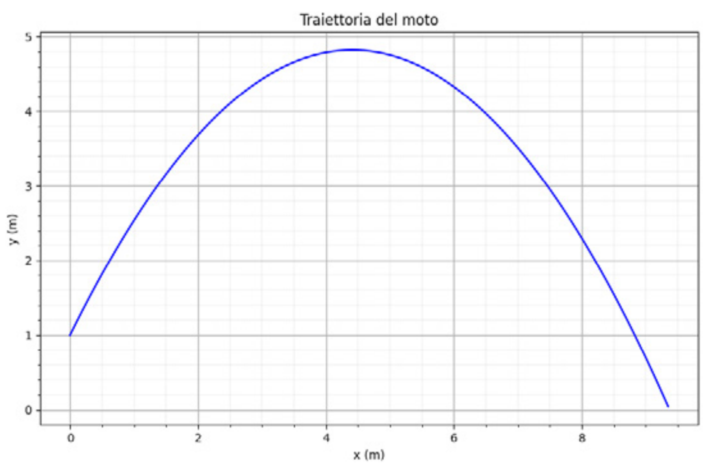
Le ulteriori modifiche, come possiamo osservare nel codice sotto, riguardano l'individuazione del tempo di volo.

```
20 #==========================================================================================
21 #EQUAZIONI DEL MOTO
22 x = v0_x*t
23 y = y0 + v0_y*t - 1/2*g*t**2
24 t_volo = y >= 0
25 #==========================================================================================
26
27 #==========================================================================================
28 #CREAZIONE DEL GRAFICO
29 fig, grafico = plt.subplots(figsize=(10,6))
30 grafico.plot(x[t_volo], y[t_volo], color='blue')
31 grafico.set_xlabel("x (m)")
32 grafico.set_ylabel("y (m)")
33 grafico.set_title("Traiettoria del moto")
34 grafico.minorticks_on()
35 grafico.grid(which='major', axis='both', linewidth=1.0)
36 grafico.grid(which='minor', axis='both', linestyle ="--", linewidth=0.2)
37 plt.show()
38 #==========================================================================================
```

Abbiamo aggiunto la riga 24 che calcola la condizione `y >= 0` e la assegna all'array di booleani `t_volo`. Tale condizione è vera quando il proiettile ha una coordinata *y* positiva e quindi è in volo.

L'ultima modifica è a riga 30, dove, facendo uso dell'indicizzazione booleana `x[t_volo]` e `y[t_volo]`, rappresentiamo la traiettoria considerando soltanto le coordinate *x* e *y* del proiettile quando è in volo.

In tal modo il grafico della traiettoria risulta troncato come nella figura sottostante.



Possiamo ora svolgere più facilmente esercizi come il seguente.

ESERCIZIO

Fissa la velocità iniziale (riga 10) e l'altezza del punto da cui è effettato il lancio (riga 9) ed esegui il programma per trovare la gittata con diversi valori dell'angolo di lancio.

- Verifica empiricamente che la gittata massima del proiettile si ha quando l'angolo di lancio vale 45°.
- Riporta i risultati in un grafico.