# PROJEKT SEMESTRALNY
# RZUT UKOŚNY

KAMIL SMARDZ

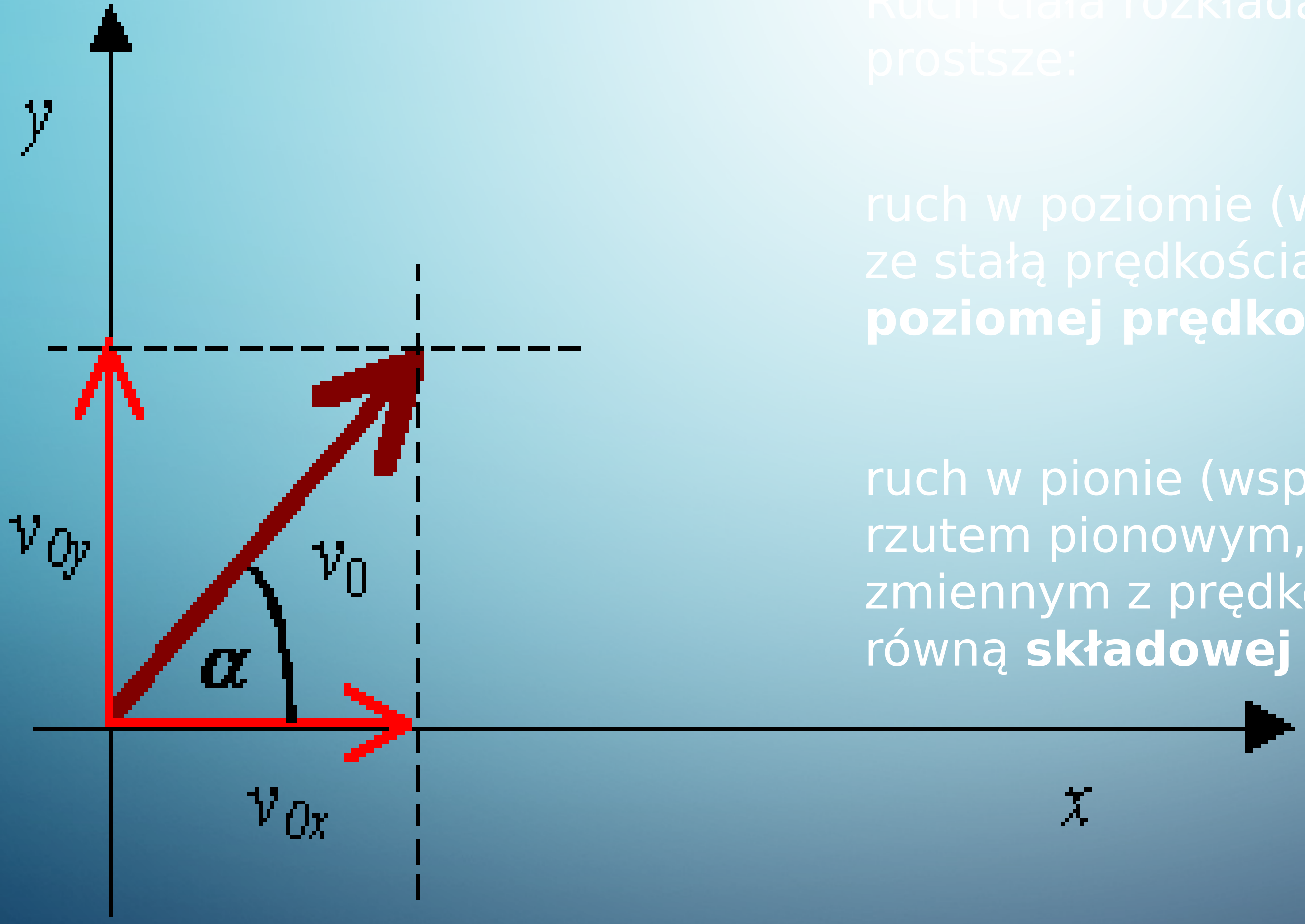
MARTYNA GABRYŚ

RAFAŁ ANTAS

Wydział Fizyki, Matematyki i Informatyki Politechnika Krakowska im. T. Kościuszki

02.01.2017

- Zadanie polegało na napisaniu programu w Pythonie demonstrującego rzut ukośny z uwzględnieniem siły grawitacji.
- Rzut ukośny – ruch w jednorodnym polu grawitacyjnym z prędkością początkową o kierunku ukośnym do kierunku pola. Ruch ten odpowiada ruchowi ciała rzuconego pod kątem do poziomu.



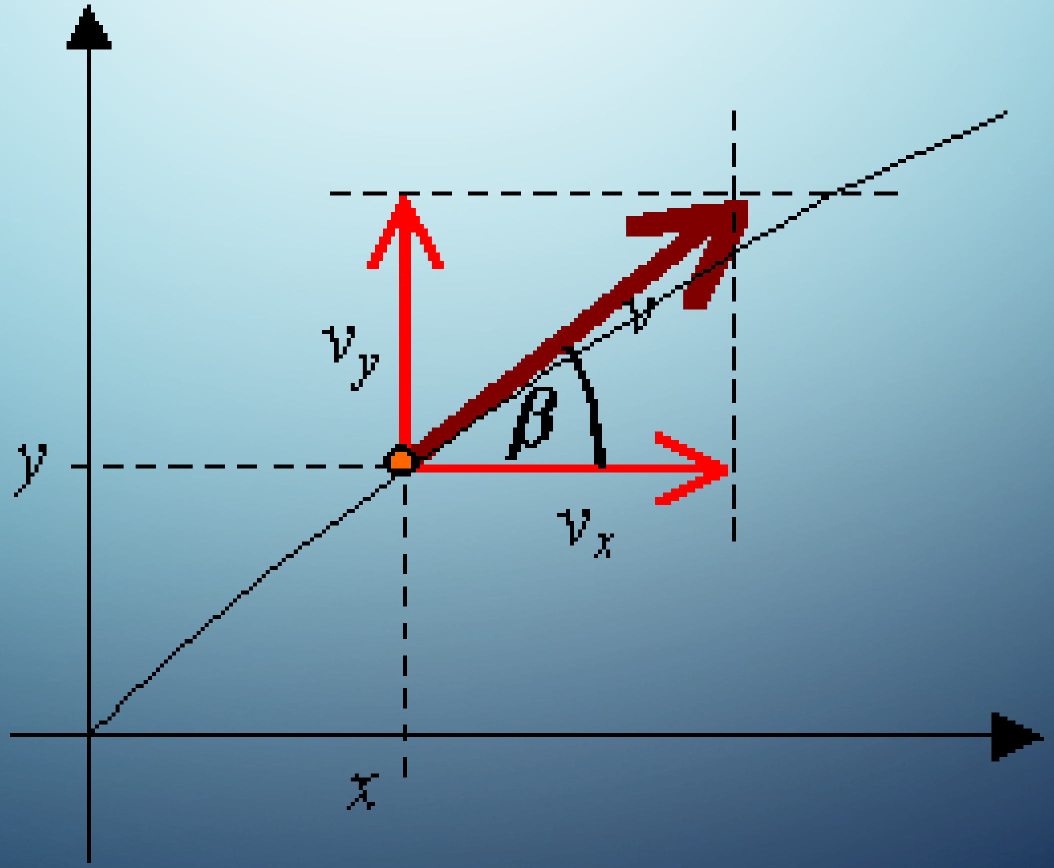

Torem ruchu ciała jest parabola.
Ruch ciała rozkłada się wtedy na dwa ruchy prostsze:

ruch w poziomie (współrzędna X-owa) – odbywa się ze stałą prędkością o wartości **składowej poziomej prędkości początkowej** $v_{0X}$

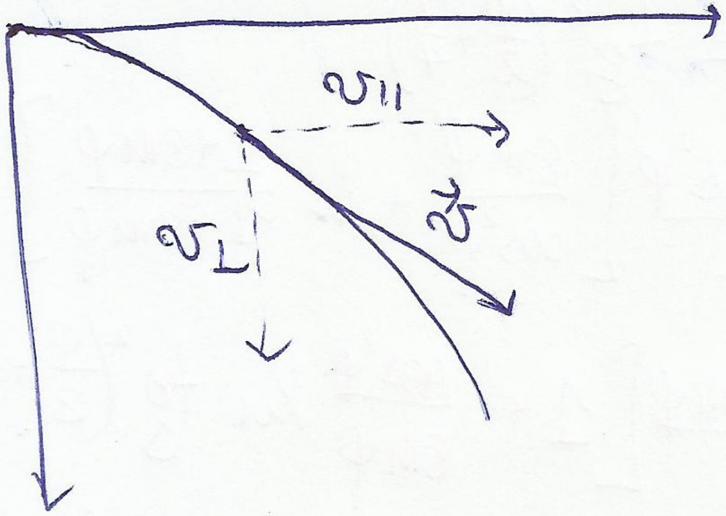
ruch w pionie (współrzędna Y-owa) – jest w istocie rzutem pionowym, czyli ruchem jednostajnie zmiennym z prędkością początkową równą **składowej pionowej** $v_{0Y}$

**Wzory opisujące rzut ukośny**
**Prędkość pozioma $v_x$ (w dowolnej chwili czasu $t$):**
$v_x = v_0 \cdot \cos \alpha$

**Prędkość pionowa $v_y$ po czasie $t$:**
$v_y = v_0 \cdot \sin \alpha - g \cdot t$



# DROGA W RZUCIE UKOŚNYM




$$\frac{ds}{dt} = v$$

$$v_{\perp} = v_0 - at$$

$$v_0 = 0$$

(praktycznie tylko prędkość w kierunku pionowym)

$$\frac{ds}{dt} = \sqrt{v_{\parallel}^2 + g^2t^2}$$

$$ds = \sqrt{v_{\parallel}^2 + g^2t^2}\, dt$$

$$s = \int_0^{t_1} \sqrt{v_{\parallel}^2 + g^2t^2}\, dt$$

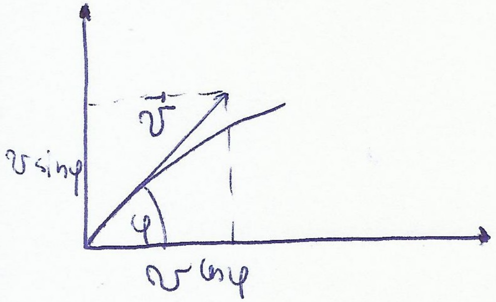
Całki typu:

$$\int\sqrt{1+x^2}\,dx = \left\{\begin{array}{ll} x = \operatorname{sh}\varphi & \operatorname{ch}^2\varphi - \operatorname{sh}^2\varphi = 1 \\ \operatorname{sh}\varphi = \dfrac{e^{\varphi} - e^{-\varphi}}{2} & \operatorname{ch}\varphi = \sqrt{1+\operatorname{sh}^2\varphi} \\ dx = \operatorname{ch}\varphi\, d\varphi & \\ \operatorname{ch}^2\varphi = \frac{1}{2} + \frac{1}{2}\operatorname{ch}2\varphi & \operatorname{sh}2\varphi = 2\operatorname{ch}\varphi\operatorname{sh}\varphi \end{array}\right\} =$$

$$= \int \operatorname{ch}^2\varphi\, d\varphi = \int \tfrac{1}{2}\, d\varphi + \int \tfrac{1}{2}\operatorname{ch}2\varphi\, d\varphi = \tfrac{1}{2}\varphi + \tfrac{1}{4}\operatorname{sh}2\varphi + c =$$

$$= \frac{1}{2}\varphi + \frac{1}{2}\operatorname{sh}\varphi\operatorname{ch}\varphi + C = \left\{\begin{array}{l} x = \dfrac{e^{\varphi} - e^{-\varphi}}{2} \\ 2x + e^{-\varphi} - e^{\varphi} = 0 \,/\cdot e^{\varphi} \\ e^{2\varphi} - 2x e^{\varphi} - 1 = 0 \\ (e^{\varphi} - x)^2 - x^2 - 1 = 0 \\ e^{\varphi} - x = \pm\sqrt{1+x^2} \\ e^{\varphi} = x \pm \sqrt{1+x^2} \\ e^{\varphi} > 0 \Rightarrow e^{\varphi} = x + \sqrt{1+x^2} \\ \varphi = \ln\left(x + \sqrt{1+x^2}\right) \end{array}\right\} =$$

$$= \frac{1}{2}\ln\left(x + \sqrt{1+x^2}\right) + \frac{1}{2}x\sqrt{1+x^2} + C$$




$$x(t) = v\cos\varphi\, t$$
$$y(t) = v\sin\varphi\, t - \tfrac{1}{2} g t^2$$
$$v_x(t) = \dot{x}(t) = v\cos\varphi$$
$$v_y(t) = \cancel{v\sin}\ \dot{y}(t) = v\sin\varphi - gt$$

$$s = \int_0^{t_1} \sqrt{v_x^2 + v_y^2}\, dt = \int_0^{t_1} \sqrt{v^2\cos^2\varphi + (v\sin\varphi - gt)^2}\, dt$$

$$\int_0^{t_1} \sqrt{v^2\cos^2\varphi + (v\sin\varphi - gt)^2}\, dt = \int_0^{t_1} v\cos\varphi \sqrt{1 + \frac{(v\sin\varphi - gt)^2}{v^2\cos^2\varphi}}\, dt =$$

$$x_1 = \frac{v\sin\varphi - g t_1}{v\cos\varphi}$$

$$= \left\{ \begin{array}{l} \dfrac{v\sin\varphi - gt}{v\cos\varphi} = x \\ -\dfrac{g}{v\cos\varphi}\, dt = dx \end{array} \right\} = -\frac{v^2\cos^2\varphi}{g} \int_{tg\varphi}^{x_1} \sqrt{1+x^2}\, dx =$$

$$= -\frac{v^2\cos^2\varphi}{2g} \Big[ \ln\left(x + \sqrt{1+x^2}\right) + x\sqrt{1+x^2} \Big]_{tg\varphi}^{x_1 = -tg\varphi}$$

$$t_1 = v\cos\varphi\, t_1 = \tfrac{1}{2} g t_1^2 \Rightarrow t_1 = \frac{2 v\sin\varphi}{g}$$

$$x_1 = -tg\varphi$$

bo $x_1 = \dfrac{v\sin\varphi - g \cdot \dfrac{2v\sin\varphi}{g}}{v\cos\varphi} = -\dfrac{v\sin\varphi}{v\cos\varphi} = -tg\varphi$

# ZASIĘG I WYSOKOŚĆ MAKSYMALNA

$$x = V_0 \cos\alpha \, t_c$$

$$t_c = 2 t_{\frac{1}{2}} = \frac{2 V_0 \sin\alpha}{g}$$

$$x = \frac{2 V_0^2 \sin\alpha \cos\alpha}{g}$$

$$Z = x = \frac{V_0^2 \sin 2\alpha}{g}$$

$$y = V_0 \sin\alpha \, t - \frac{g t^2}{2}$$

$$V_y = V_0 \sin\alpha - g t$$

$$V_y = 0 \Leftrightarrow t_{\frac{1}{2}} = \frac{V_0 \sin\alpha}{g}$$

$$y = \frac{V_0^2 \sin^2\alpha}{g} - \frac{V_0^2 \sin^2\alpha}{2g}$$

$$y = H_{max} = \frac{V_0^2 \sin^2\alpha}{2g}$$

# ANALIZA KODU

Ta część kodu pozwala nam na wprowadzenie dowolnych danych potrzebnych do wykonania zadania

```
#Predkosc pocisku u wylotu lufy (w m/s)
predo = raw_input("Prędkość Vo pocisku u wylotu lufy: ")
predo = float(predo)
#Kąt wylotu pocisku (w stopniach)
kat = raw_input("kat: ")
kat = float(kat)
#Odleglosc miedzy dzialem a celem w metrach
odleglosc = raw_input("Odległość: ")
odleglosc = float(odleglosc)
```

Funkcja raw_input wyświetla string, który w dalszej część kodu zmienia się na float

# Dana część kodu pozwala na stworzenie przestrzeni w której pocisk będzie się poruszać

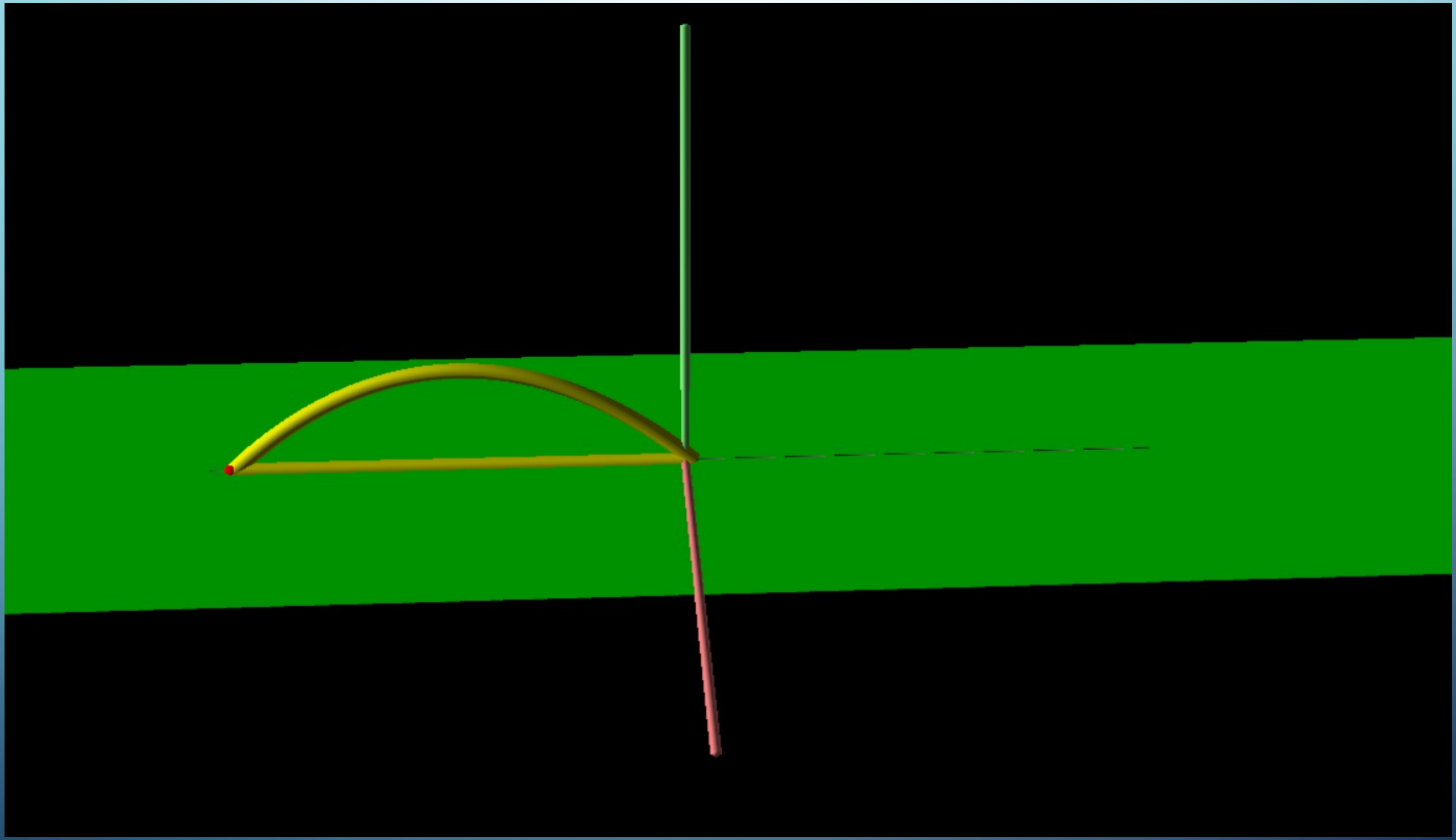
```
# Osie
osx = curve(pos=[(0,0,0), (L,0,0)],radius=10, color=(1.5,0.5,0.5))
osy = curve(pos=[(0,0,0), (0,L,0)],radius=10, color=(0.5,1.5,0.5))
osz = curve(pos=[(0,0,-L), (0,0,L)], color=(0.5,0.5,0.5))
# Trawa
trawa = box(pos=(0,0,0), length=odleglosc+100, height=1, width=odleglosc+5000, color=color.green)
# Punkt startowy
cel = sphere(pos=(0,0,odleglosc), radius=20, color=color.red)
# Pocisk
pocisk = sphere(pos=(0,0,0), radius=10, color=color.blue)
# Tor pocisku
pocisk.trail = curve(pos=[pocisk.pos], color=color.yellow, radius=20)
```

Podany fragment kodu przedstawia pętlę while, która wykonuje się dopóki warunek if nie zostanie spełniony

```
# Petla animacyjna
t = 0
while 1:
    rate(500)
    t = t + 0.005
    pocisk.pos.x = 0
    pocisk.pos.y = predo*t*sin(kat)- 0.5*g*t**2
    pocisk.pos.z = odleglosc-predo*t*cos(kat)
    pocisk.trail.append(pos=pocisk.pos)

    if (pocisk.y <= 0):
        print "Pocisk wyladowal w punkcie (%f, %f, %f)" % (pocisk.pos.x, pocisk.pos.y, pocisk.pos.z)
        break # Zatrzymaj petle, aby program zakonczyl dzialanie
```

# WYNIKI PROGRAMU



# PODSUMOWANIE

Program spełnia swoją funkcję, którą było narysowanie wykresu rzutu ukośnego, wyliczenie zasięgu, maksymalnej wysokości, drogi kulki. Program można rozwinąć, dołączyć do niego siłę oporu.