

Programski alati u fizici

Ovisnost dometa projektila o koeficijentu otpora zraka i masi projektila

Iris Butigan

Split, 9.6.2023.

Sažetak

Cij ovoga rada je ispitati ovisnost dometa projektila u ovisnosti o koeficijentu otpora zraka C_d te o masi projektila m . U svrhu ovoga rada razvijen je kod koji sadrži klasu *Projectile* koja ima implementirane metode za simuliranje kosog hitca u dvije dimenzije s otporom zraka koristeći Runge-Kutta metodu četvrtog reda. Kod je napisan u programskom jeziku Python koristeći biblioteke NumPy i Matplotlib. Koristeći napisani kod i ispitivanjem ovisnosti dometa o navedenim veličinama, dolazi se do zaključka kako domet projektila raste povećanjem mase, dok povećanjem koeficijenta trenja zraka opada.

1 Uvod

Kosi hitac jedan je od fundamentalnih koncepata u fizici, koji se koristi za proučavanje gibanja projektila. Kada se projektil giba u zraku, suočava se s otporom zraka koji može značajno utjecati na njegovu putanju. Početni parametri relevantni za proučavanje gibanja projektila su masa m , početna brzina v_0 te kut θ pod kojim je projektil ispaljen u odnosu na horizontalnu os. Polazeći od principa Newtonove mehanike vrijedi drugi Newtonov zakon:

$$\sum_i \vec{F}_i = m\vec{a}$$

Sile koje djeluju na projektil su gravitacijska sila \vec{F}_g i sila otpora zraka \vec{F}_o pa gornja jednadžba postaje $\sum_i \vec{F}_i = m\vec{a} = \vec{F}_g + \vec{F}_o$. Gravitacijska sila u svakom trenutku djeluje vertikalno prema dolje, dok je sila otpora zraka ovisna o brzini projektila v u nekom trenutku te uvijek djeluje suprotno od smjera gibanja čestice. Sila otpora zraka ovisi još i o površini tijela okomitoj na smjer gibanja A , koeficijentu trenja zraka C_d i gustoći zraka ρ . Sada imamo

$$m\vec{a} = \vec{F}_g + \vec{F}_o = m\vec{g} - \frac{\rho C_d A}{2} v^2 \hat{v}$$

iz čega slijedi

$$\vec{a} = \frac{d\vec{v}(t)}{dt} = \vec{g} - \frac{\rho C_d A}{2m} v^2 \hat{v}$$

Rastavimo sada gibanje na horizontalne (x) i vertikalne (y) komponente:

$$a_x(t) = -\operatorname{sgn}(v_x) \frac{\rho C_d A}{2m} v_x^2, a_y(t) = -g - \operatorname{sgn}(v_y) \frac{\rho C_d A}{2m} v_y^2$$

$$v_x(t) = v_{0x} + a_x t, v_y(t) = v_{0y} + a_y t$$

$$x(t) = x_0 + v_x t, y(t) = y_0 + v_y t$$

pri čemu je $v_{0x} = v_0 \cos \theta$ i $v_{0y} = v_0 \sin \theta$.

2 Diskusija i rezultati

Za rješavanje ovog problema u Pythonu odabrali smo Runge-Kutta metodu četvrtog reda naspram Euloreve zbog veće preciznosti Runge-Kutta metode. Eulerova metoda daje numerički stabilna i precizna rješenja samo za jako male vremenske korake dt . S druge strane, Runge-Kutta metoda ostvaruje veću ili jednaku preciznost kao Eulerova metoda, ali obavi manje računskih operacija te je zbog toga ekonomičnija za izvršavanje koda. Formule za primjenu Runge-Kutta metode četvrtog reda zadane su na sljedeći način:

$$\begin{aligned} k_{1v} &= a(x^n, v^n, t^n) dt, \quad k_1 = v^n dt \\ k_{2v} &= a\left(x^n + \frac{k_1}{2}, v^n + \frac{k_1}{2}, t^n + \frac{k_1}{2}\right) dt, \quad k_2 = \left(v^n + \frac{k_{1v}}{2}\right) dt \\ k_{3v} &= a\left(x^n + \frac{k_2}{2}, v^n + \frac{k_2}{2}, t^n + \frac{k_2}{2}\right) dt, \quad k_3 = \left(v^n + \frac{k_{2v}}{2}\right) dt \\ k_{4v} &= a\left(x^n + \frac{k_3}{2}, v^n + \frac{k_3}{2}, t^n + \frac{k_3}{2}\right) dt, \quad k_4 = \left(v^n + \frac{k_{3v}}{2}\right) dt \\ v^{n+1} &= v^n + \frac{1}{6}(k_{1v} + 2k_{2v} + 2k_{3v} + k_{4v}) \\ x^{n+1} &= x^n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4). \end{aligned}$$

Runge-Kutta metoda u kodu je napisana kako se vidi na Slici 1 uz pomoćnu metodu $\mathbf{a}(\mathbf{v})$ koja računa koeficijent $a(x, v, t)$ po parametru v i podatka r koji je definiran kao $r = \frac{\rho C_d A}{2m}$. Metoda `_move_rk4()` se primjenjuje sve dok projektil ne pogodi tlo ($y = 0$).

```

26 def a(self, v):
27     av = -r * sign(v) * self.r * (v**2)
28     return av
29
30 def _move_rk4(self):
31     k1v = self.a(self.v[1]) * self.dt
32     k1x = self.v[1] * self.dt
33     k2v = self.a((self.v[1] + k1v/2)) * self.dt
34     k2x = (self.v[1] + k1v/2) * self.dt
35     k3v = self.a((self.v[1] + k2v/2)) * self.dt
36     k3x = (self.v[1] + k2v/2) * self.dt
37     k4v = self.a((self.v[1] + k3v/2)) * self.dt
38     k4x = (self.v[1] + k3v/2) * self.dt
39
40     self.v.append(self.v[1] + (k1v + 2*k2v + 2*k3v + k4v)/6)
41     self.x.append(self.x[1] + (k1x + 2*k2x + 2*k3x + k4x)/6)
42
43     k1y = (g + self.a(self.v[1])) * self.dt
44     k1y = self.v[1] * self.dt
45     k2y = (g + self.a((self.v[1] + k1y/2))) * self.dt
46     k2y = (self.v[1] + k1y/2) * self.dt
47     k3y = (g + self.a((self.v[1] + k2y/2))) * self.dt
48     k3y = (self.v[1] + k2y/2) * self.dt
49     k4y = (g + self.a((self.v[1] + k3y/2))) * self.dt
50     k4y = (self.v[1] + k3y/2) * self.dt
51
52     self.v.append(self.v[1] + (k1y + 2*k2y + 2*k3y + k4y)/6)
53     self.y.append(self.y[1] + (k1y + 2*k2y + 2*k3y + k4y)/6)
54
55     self.t.append(self.t[1] + self.dt)
56

```

Figure 1: Snimka zaslona dijela koda: metoda `_move_rk4()` koja implementira Runge-Kutta metodu za numeričko rješavanje problema i pomoćna metoda $\mathbf{a}(\mathbf{v})$

U programu *seminar.py* pozvana je klasa *Projectile*. U svrhu ovoga rada promatramo domet projektila u ovisnosti o koeficijentu otpora zraka C_d te o masi projektila m . Vrijednosti koeficijenta otpora zraka za koje promatramo domet kreću se u rasponu od 0.1 do 0.5, dok se vrijednosti promatranih masa m kreću od 1 kg do 10 kg. Primjenom razvijenog koda na ove veličine te vizualizacijom dobivenih podataka koristeći *matplotlib.pyplot* dobivamo bolji uvid u ovisnost dometa koeficijentu trenja, odnosno masi. Uvidom u do-

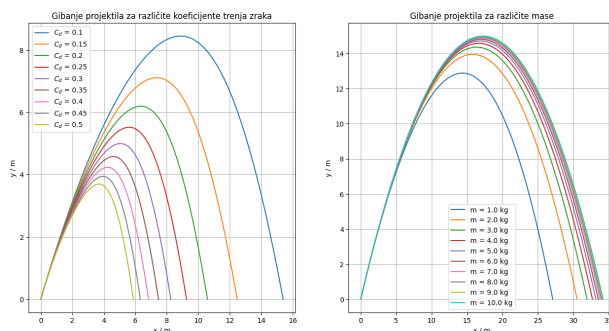


Figure 2: x-y grafovi: Grafički prikaz ovisnosti dometa o koeficijentu trenja zraka (lijevo) i grafički prikaz ovisnosti dometa o masi (desno).

bivene grafove lako se vidi kako domet ovisi o promatranim veličinama. Povećanjem mase projektila, domet se također povećava. S druge strane, povećanjem koeficijenta trenja zraka domet se smanjuje. S obzirom na to da veća masa znači veću silu, prirodno je očekivati kako će u slučaju veće mase i domet biti veći. Prikazani graf potvrđuje ovakvu ovisnost mase i dometa. Sila otpora zraka suprotstavlja se gibanju projektila zbog čega ga usporava te se postiže manji domet nego u slučaju kada nema otpora zraka. Sila otpora zraka proporcionalna je koeficijentu trenja zraka, te se sukladno tome smanjenjem koeficijenta trenja smanjuje i sila otpora zraka te su efekti otpora zraka sve slabiji. Za vrlo sitne vrijednosti koeficijenata trenja zraka može se zanemariti utjecaj sile otpora zraka.

Grafičkim prikazom različitih koeficijenata trenja zraka na x -osi i odgovarajućih dometa na y -osi primjećuje se padajuća eksponencijalna ovisnost ovih veličina. Bilo je i očekivano da će ovisnost biti padajuća. Na isti način je i prikazana ovisnost dometa o masi, no u tom je slučaju ovisnost rastuća i logaritamska, odnosno domet logaritamski raste povećanjem mase projektila. Također je potvrđeno očekivanje o rastu dometa povećanjem mase.

Kod razvijen u svrhu ovoga rada temelji se na sedmoj vježbi kolegija Programski alati u fizici. Klasa *Projectile* definirana je i inicijalizirana metodom `__init__` koja koristi zadane ulazne parametre za daljni razvoj koda te definira potrebna polja koja kasnije u kodu spremaju fizikalne veličine potrebne za numeričko rješavanje problema. Privatna metoda `__move_euler` kreirana je u svrhu numeričkog rješavanja kinematičkih jednadžbi u kratkom vremenskom koraku dt . Smanjenjem vremenskog koraka dt postiže se rezultat bliži pravom fizikalnom rješenju problema, odnosno smanjenjem koraka kod postiže veću preciznost. Međutim, Eulerova metoda nije najpreciznija zbog čega se u ovom radu koristi Runge-Kutta metoda četvrtog reda kako je već prethodno opisano. Runge-Kutta metoda u kodu je definirana metodom `__move_rk4` koja je prikazana na Slici 1. Uz tu je metodu prethodno definirana pomoćna metoda $a(v)$ koja kao ulazni parametar prima brzinu v te vraća akceleraciju potrebnu za izračun unutra Runge-Kutta metode.

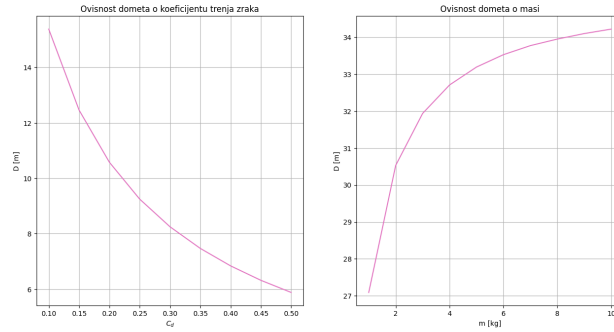


Figure 3: Ovisnost dometa projektila o koeficijentu trenja zraka (lijevo) i ovisnost dometa projektila o masi (desno) s pripadajućim krivuljama regresije.

Ova je pomoćna metoda definirana kako bi kod bio jednostavniji i čistiji. Gibanje projektila definirano je metodom *gibanje_rk4* koja poziva Runge-Kutta metodu te računa kinematičke veličine sve dok projektil ne udari o tlo ($y \geq 0$), odnosno dok y udaljenost ne postigne negativnu vrijednost. U simulacijama provedenim u svrhu ovoga rada svi su parametri osim koeficijenta trenja zraka u prvom i mase projektila u drugom slučaju bili fiksni te su iznosili: $v_0 = 20 \text{ m s}^{-1}$, $\theta = 60^\circ$, $dt = 0.001 \text{ s}$, $\rho = 1.29 \text{ kg m}^{-3}$, $A = 0.1 \text{ m}^2$

3 Zaključak

Razvojem ovoga koda numerički je riješen problem gibanja projektila u $x - y$ ravnini uzevši u obzir otpor zraka. Simuliranjem gibanja za različite koeficijente trenja zraka uočeno je kako domet projektila eksponencijalno opada povećanjem koeficijenta trenja. Iz simulacija gibanja za različite mase zaključuje kako domet raste povećanjem mase. Dobiveni su rezultati u skladu s predviđanjima. Kod je relativno precizan posebice za vrlo male vremeske korake, ali bi se mogao poboljšati korištenjem neke naprednije i preciznije metode od Runge-Kutta metode koja je sama po sebi vrlo precizna.

References

- [1] A. Dulčić, *Mehanika*
- [2] T. Šćulac, *Materijali s predavanja*