

Simulační studie

9. Balistika ve vojenství

Modelování a simulace

Varianta 1: Trajektorie střely (dělostřelecké, raketové)

5. prosince 2022

Ivan Golikov (xgolik00)
Maksym Koval (xkoval20)

Obsah

1	Úvod a motivaci	1
1.1	Autori, zdroje	1
1.2	Ověřování validity modelu	1
2	Fakta	1
2.1	Použité technologie	1
2.1.1	C++ a make	1
2.1.2	Python	1
2.2	Původ použitých technologií	1
3	Koncepce modelu	2
3.1	Popis konceptuálního modelu	2
4	Architektura simulačního modelu	3
4.1	Spouštění simulačního modelu	4
5	Experimentování	5
6	Závěr	7

1 Úvod a motivaci

Cílem projektu je simulovat trajektorii střely (vypuštěné z děla nebo houfnice) za podmínek blízkých skutečným. Na základě simulace [8, snímek 8] ukážeme vliv odporu vzduchu na trajektorii střely. Téma bylo pro nás nejzajímavější z témat "vojenských simulací". Ve své práci jsme se chtěli přiblížit skutečným vojenským simulacím a pochopit problematiku tohoto tématu. Rádi bychom se spojili s odborníkem z této oblasti, ale bohužel jsme takové kontakty nenašli.

1.1 Autori, zdroje

Nad projektem pracovali studenti VUT FIT Koval Maksym a Golikov Ivan. Jako zdroje byly použity knihy o balistice a mechanice. Byla také použita řada zdrojů zdrojů jiných autorů, které jsou veřejně dostupné. Všechny použité zdroje jsou uvedeny v citaci.

1.2 Ověřování validity modelu

Jako ověření správnosti modelu byly použity programy napsané jinými uživateli z internetu, jejichž metody řešení se lišily od řešení uvedeného v projektu. Všechny zdroje použité k ověření modelu jsou uvedeny v citaci.

2 Fakta

Naše téma zahrnovalo simulaci trajektorie střely z děla nebo houfnice. Abychom mohli vypočítat trajektorii střely, potřebujeme takové základní pojmy fyziky, jako je gravitace, počáteční rychlost objektu, hmotnost objektu, úhel výstřelu. Aby bylo počítání co nejbližší skutečnému, stojí za zvážení také takové hodnoty jako: rychlost a směr větru, odpor prostředí (v našem případě vzduchu), teplota prostředí (srážky). Tvar objektu ovlivňuje koeficient odporu objektu [4, drag coefficient], který také významně ovlivňuje trajektorii objektu. V našich výpočtech nebudeme počítat odpor objektu osobně, ale vezmeme jej ze zdroje Wikipedia [4, drag coefficient].

Údaje o schválení můžeme potvrdit na stránce Wikipedie o balistice. [5].

2.1 Použité technologie

2.1.1 C++ a make

K řešení matematické stránky projektu jsme použili programovací jazyk C++, ve standardu C++14. Pro zjednodušení sestavení programu byl také použit nástroj GNU make [6].

2.1.2 Python

Pro vizualizaci dat jsme použili programovací jazyk Python ve verzi 3.8.10. a knihovnu pro vykreslování pro Python matplotlib [7].

2.2 Původ použitých technologií

Programovací jazyk C++ byl použit protože je vhodný pro rychlé algoritmické výpočty, má objektno orientovaný přístup a pomocné nástroje ve formě knihoven pro práci s vektory a dalšími datovými strukturami. Python a knihovna matplotlib byla vybrána námi na základě zkušeností z předchozích projektů. Knihovna matplotlib je vhodná pro vizualizaci statistik a diskrétních datových typů.

3 Koncepce modelu

V této části budeme analyzovat a formalizovat koncept modelu. Uvedeme také fakta, která jsme zanedbali. Tak, jak jsme již uvedli v sekci "Fakta", abychom vytvořili trajektorii, potřebujeme:

1. Gravitační sílu.
2. Počáteční rychlost objektu.
3. Úhel počáteční trajektorie.
4. Odpor objektu.
5. Hmotnost objektu.
6. Plochu průřezu objektu.
7. Odpor okolí.
8. Okolní teplotu.

Rozhodli jsme se zanedbat okolní teplotu, protože má velmi malý vliv na konečný výsledek simulace [1]. Pro parametr odporu jsme použili veřejně dostupná fakta o odporu objektů z otevřených zdrojů, protože výpočet těchto dat by se mohl rozdělit na další práci [4]. Ze všech výše uvedených vstupů jsme schopni vypočítat všechny síly působící na objekt a také vytvořit diferenciální rovnice pro nalezení bodů, na kterých leží trajektorie vypuštěného objektu, v našem případě projektilu děla.

3.1 Popis konceptuálního modelu

Nejprve tedy budeme muset zjistit sílu, která působí na objekt v důsledku gravitace, tíhu. K tomu potřebujeme jeho hmotnost a konstantu g .

$$G = -mg$$

Hodnoty vektorů rychlosti můžeme vypočítat na základě úhlu a počáteční rychlosti výstřelu:

$$v_x = v \cos \alpha$$

$$v_y = v \sin \alpha$$

Sílu odporu můžeme spočítat pomocí vzorce:

$$D = 1/2 r_a v^2 c_d A$$

Kde r_a je hustota prostředí, c_d koeficient odporu objektu, A je plocha jeho průřezu.

Složky odporu můžeme získat tímto způsobem:

$$D_x = -Dv_x$$

$$D_y = -Dv_y$$

Konečné složky odporových sil označíme jako F:

$$F_x = D_x$$

$$F_y = G + D_y$$

Vydělením konečných odporových sil hmotností objektu získáme jeho zrychlení:

$$F_x/m = d_x^2/d_t^2$$

$$F_y/m = d_y^2/d_t^2$$

Získali jsme sadu dvou řadových diferenciálních rovnic druhého řádu s počátečními podmínkami. Existuje numerická metoda zvaná Nyströмова modifikace čtvrtého řádu Runge-Kutta metoda [3], která řeší tento druh problémů. Metoda tvrdí, že:

$$y_{n+1} = y_n + h(d_y/d_t)_n + h^2/6(k_1 + k_2 + k_3)$$

$$(d_y/d_t)_{n+1} = (d_y/d_t)_n + h/6(k_1 + 2k_2 + 2k_3 + k_4)$$

kde:

$$k_1 = f(t_n, y_n, (d_y/d_t)_n)$$

$$k_2 = f(t_n + 1/2h, y_n + 1/2h(d_y/d_t)_n + h^2/8k_1, (d_y/d_t)_n + h/2k_1)$$

$$k_3 = f(t_n + 1/2h, y_n + 1/2h(d_y/d_t)_n + h^2/8k_1, (d_y/d_t)_n + 1/2k_1)$$

$$k_4 = f(t_n + h, y_n + h(d_y/d_t)_n, (d_y/d_t)_n + hk_3)$$

Tímto způsobem jsme schopni vypočítat polohu našeho objektu pomocí kroků. Každý následující krok bude obsahovat výsledky předchozích kroků.

4 Architektura simulačního modelu

Po spuštění našeho programu s uvedením všech potřebných parametrů bude vygenerován soubor out.txt který bude obsahovat data potřebná pro sestavení grafu. Graf vytvoří soubor drawer.py.

Třída Point je datová struktura opakující vlastnost vektoru.

Třída "Flying Ball" je třída projektilu které spustíme. K jeho vytvoření budete potřebovat všechny vstupy uvedené v sekci "Koncepte modelu".

Nejdůležitější a nejzajímavější třídou je třída "Nystrom" je iterátorem a provádí základní výpočty metodou uvedenou v sekci "Popis konceptuálního modelu". Tato třída má hlavní metodu-krok který budeme spouštět ve smyčce změnou parametru času.

Třída "Wind" je potřebná pro simulaci provozu větru.

Také máme pomocnou třídu pro testování "Test".

4.1 Spouštění simulačního modelu

Simulační model je před jeho prvním spuštěním nejdříve potřeba přeložit příkazem `make`. Poté musíte program spustit pomocí příkazu `./make [ARGS]`, ale pro jeho spuštění budete muset zadat určité argumenty:

1. `-mx` - vektor pohybu objektu osy X
2. `-my` - vektor pohybu objektu osy Y
3. `-ma` - úhel, pod kterým byl předmět vypálen
4. `-v` - rychlost objektu
5. `-cd` - koeficient odporu vzduchu
6. `-wx` - vektor pohybu větru podél osy X
7. `-wy` - vektor pohybu větru podél osy Y
8. `-wz` - vektor pohybu větru podél osy Z
9. `-wf` - rychlost větru
10. `-d` - průměr objektu
11. `-ro` - hustota objektu

Pro usnadnění jsme vytvořili předem napsané testy, které jsou popsány v kapitole ???. Poté jsou všechny výsledky vloženy do souboru `out.txt` ve formátu `time 'umístění objektu na ose x': 'umístění objektu na ose y': 'umístění objektu na ose z' 'rychlost objektu na ose x': 'rychlost objektu na ose y': 'rychlost objektu na ose z'`.

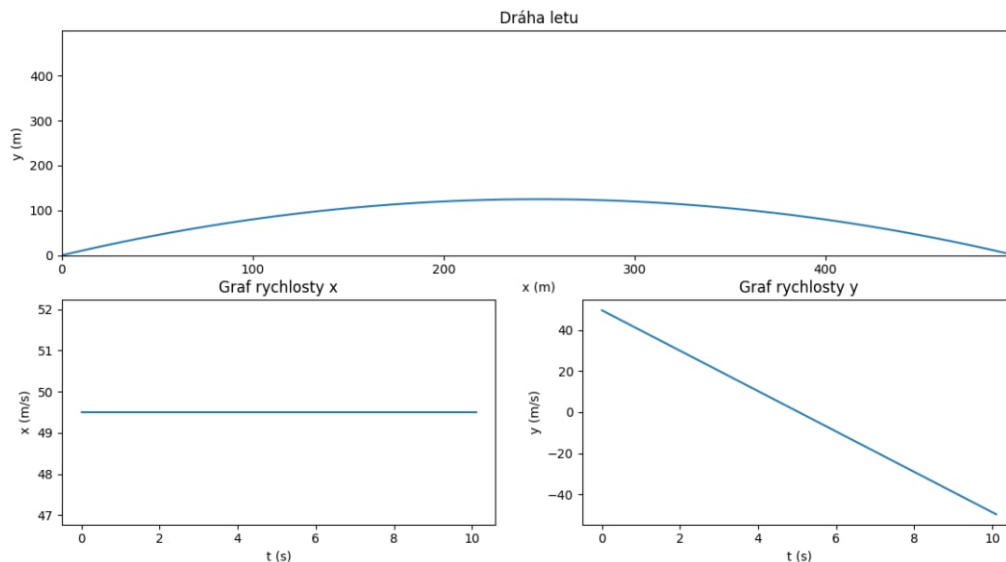
Chcete-li tato data zobrazit ve formě grafu, musíte použít náš soubor napsaný v python `drawer.py`. Chcete-li to provést, napište příkaz: `python3 drawer.py`. Poté se zobrazí graf vzdáleností podél os X a Y, změna rychlosti X v závislosti na čase a změna rychlosti Y v závislosti na čase.

5 Experimentování

Máme pět různých testů pro testování různých situací:

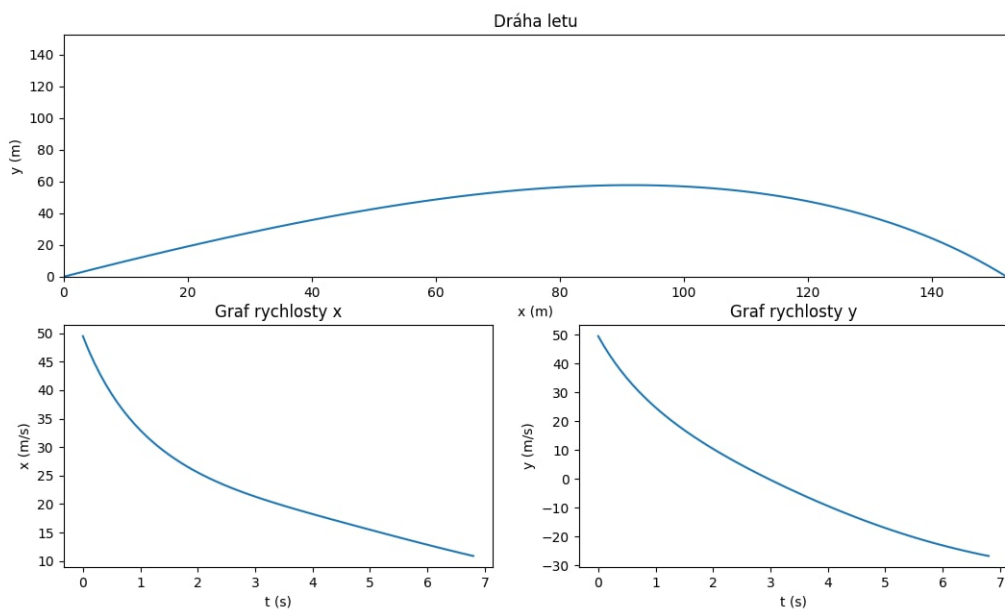
1. param: $-mx\ 1\ -my\ 0\ -ma\ 45\ -v\ 70\ -cd\ 0\ -wx\ 0\ -wy\ 0\ -wz\ 0\ -wf\ 0\ -d\ 600\ -ro\ 0.1$

cíl: Zjistěte trajektorii předmětu na plátně bez vlivu dalších sil (drag = 0,0, bezvětrí).



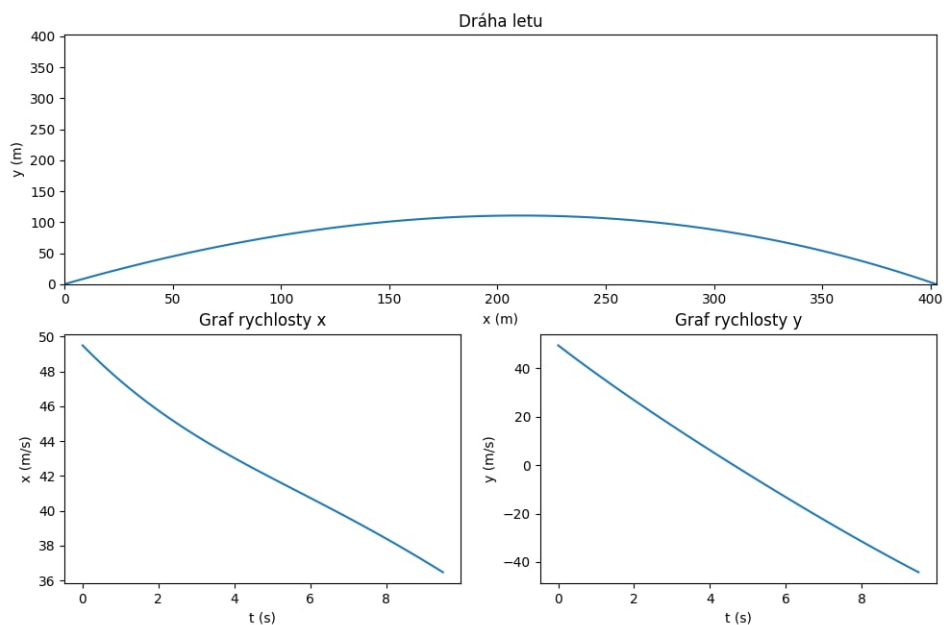
2. param: $-mx\ 1\ -my\ 0\ -ma\ 45\ -v\ 70\ -cd\ 0.47\ -wx\ 0\ -wy\ 0\ -wz\ 0\ -wf\ 0\ -d\ 600\ -ro\ 0.1$

cíl: Vliv odporu vzduchu na trajektorii objektu (drag = 0,47, bezvětrí).



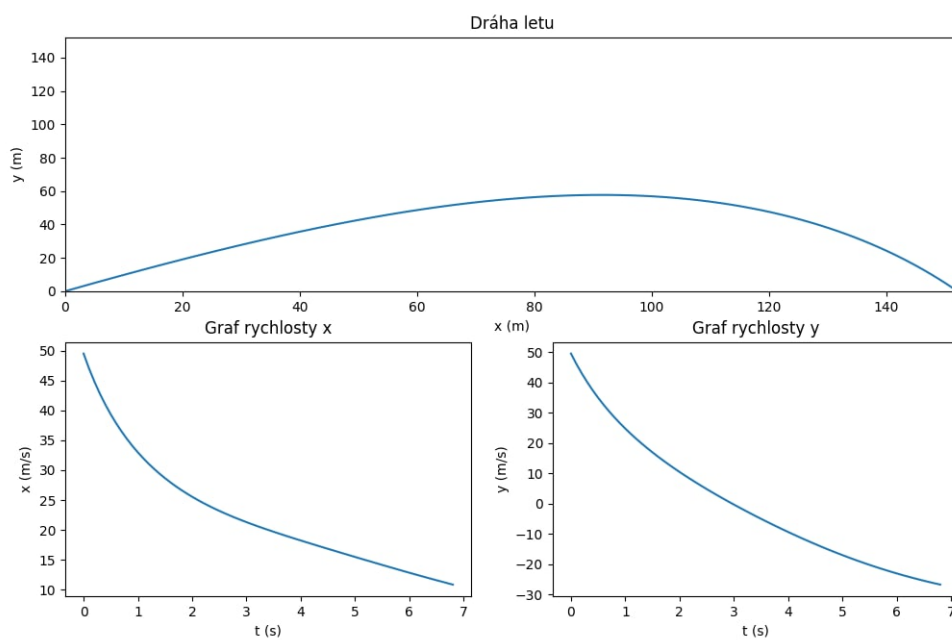
3. param: -mx 1 -my 0 -ma 45 -v 70 -cd 0.5 -wx 0 -wy 0 -wz 0 -wf 0 -d 600 -ro 0.1

cíl: V tomto experimentu jsme snížili koeficient CP, abychom zjistili, jak moc ovlivňuje trajektorii (drag = 0,04, bezvětrí).



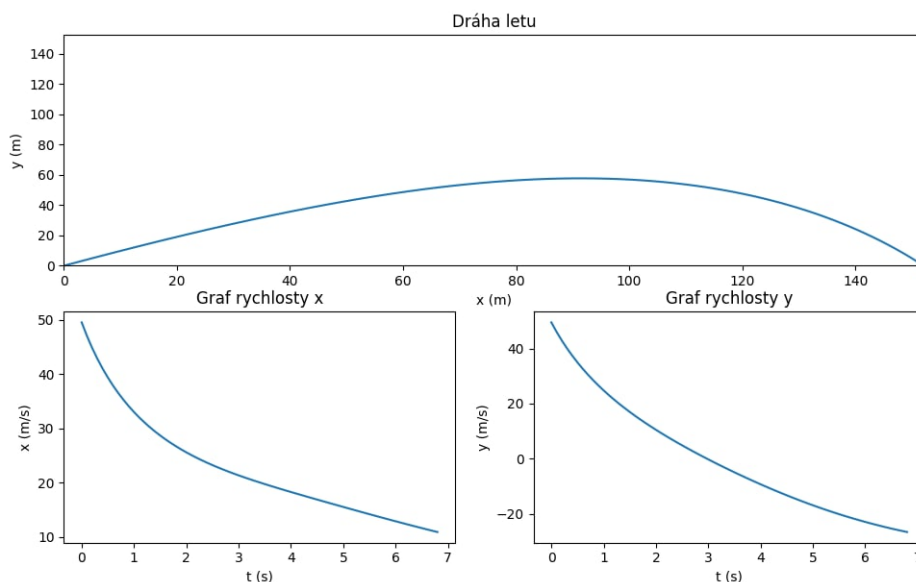
4. param: -mx 1 -my 0 -ma 45 -v 70 -cd 0.04 -wx 1 -wy 0 -wz 0 -wf 5 -d 600 -ro 0.1

cíl: V tomto experimentu se již objevuje vítr, který se pohybuje stejným směrem jako náš objekt (drag = 0,47, wind = 5 m/s).



5. param: -mx 1 -my 0 -ma 45 -v 70 -cd 0.47 -wx -1 -wy 0 -wz 0 -wf 5 -d 600 -ro 0.1

cíl: V tomto experimentu již existuje vítr, který se pohybuje opačným směrem než náš objekt (drag = 0,47, wind = -5 m/s).



Výsledky experimentů jsme porovnali s kalkulačkou na webu NASA [2].

6 Závěr

V rámci projektu jsme napsali program který vypočítá balistickou dráhu letu střely s přihlédnutím ke všem základním vlastnostem fyziky. Výsledkem bylo získání výsledků blízkých skutečným.

Část projektu, která byla zodpovědná za výpočty byla realizována v jazyce C++.

Pro vizualizaci byl použit skript Pythonu.

Odkazy

- [1] *Ball Flight Calculation Table*, Nasa. [online]. URL: <https://www1.grc.nasa.gov/beginners-guide-to-aeronautics/fly-ball-results/>.
- [2] *Ballistic Flight Calculator*, Nasa. [online]. URL: <https://www1.grc.nasa.gov/beginners-guide-to-aeronautics/fltcalc/>.
- [3] *Dormand, J.R., Prince, P.J. New Runge-Kutta algorithms for numerical simulation in dynamical astronomy*. [online]. URL: <https://doi.org/10.1007/BF01230162>.
- [4] *Drag coefficient*. [online]. URL: https://en.wikipedia.org/wiki/Drag_coefficient.
- [5] *External ballistics*. [online]. URL: https://en.wikipedia.org/wiki/External_ballistics.
- [6] *GNU make*. [online]. URL: <https://www.gnu.org/software/make/>.
- [7] *Knihovna vykreslování grafů Matplotlib*. [online]. URL: <https://matplotlib.org/>.
- [8] Petr Peringer a Martin Hruby. *Modelování a simulace*. [online]. [vid. 2022-09-15]. URL: <https://www.fit.vutbr.cz/study/courses/IMS/public/prednasky/IMS.pdf>.