

Program Výstřel z kanónu

Jiří Zeman, 2. ročník Bc. Obecná matematika

28. února 2016

1 Stručný popis

Program simuluje šikmý vrh kulového tělesa při zadaných vnějších podmínkách, jako je hustota prostředí nebo tíhové zrychlení.

2 Fyzikální model

Pro zjištění rychlosti a polohy tělesa v daném čase se používá Druhý Newtonův zákon:

$$m \frac{dv}{dt} = \sum F_i \quad (1)$$

Úloha obecně vede na soustavu dvou diferenciálních rovnic pro neznámé složky v_x a v_y rychlosti v . Obtížnost řešení takové soustavy se různí. Záleží na tom, které síly do výslednice na pravé straně rovnice (1) zahrneme. V případě tohoto programu je uvažována tíhová síla $F_g = mg$ mířící svisle dolů a odporová síla F_o působící proti směru okamžité rychlosti.

2.1 Odporová síla při malých rychlostech

Když se kulové těleso pohybuje v tekutině (kapalině nebo plynu) nízkou rychlostí (jak moc nízkou se odhaduje pomocí tzv. Reynoldsova čísla), například se zvolna potápí nebo si razí cestu v želé, platí *Stokesův zákon*:

$$F_o = 6\pi\mu Rv \quad (2)$$

kde μ je *dynamická viskozita* prostředí a R poloměr tělesa. Odporová síla je tedy úměrná první mocnině rychlosti v a separací proměnných lze vypočítat rychlost v

$$v_x(t) = v_{0x} e^{-\frac{b}{m}t} \quad (3)$$

$$v_y(t) = -\frac{mg}{b} + (v_{0y} + \frac{mg}{b})e^{-\frac{b}{m}t} \quad (4)$$

Pro zkrácení zápisu kladu $b = 6\pi\mu$. Koule je tekutinou **laminárně obtékána**, nevznikají víry.

2.2 Odporová síla při vyšších rychlostech

Při vyšších rychlostech proudí kapalina v okolí tělesa **turbulentně** a odporovou sílu se standardně popisuje Newtonovým odporovým vzorcem

$$F_o = \frac{1}{2}CS\rho_{prost}v^2 \quad (5)$$

v němž C se nazývá součinitel odporu (pro kouli má hodnotu $\frac{1}{2}$), S znamená plochu příčného průřezu tělesa a ρ_{prost} označuje hustotu prostředí. Závislost odporové síly na kvadrátu rychlosti vnáší do problému nelinearitu a rovnici už nelze snadno vyřešit.

2.3 Odporová síla v obecném případě

Platnost vztahů (2) a (5) má své meze a v některých případech je přesnější považovat odporovou sílu za úměrnou jiné, neceločíselné, mocnině rychlosti. Příklad

$$F_o = bv^\alpha, \quad \alpha \in (1.75, 3) \quad (6)$$

kdy čísla b, α jsou určena měřením, jsem si vyzkoušel modelovat v tomto programu. Takovou odporovou sílu zde pracovně nazývám *empirickou*, protože její vyjádření neplyne z obecných fyzikálních zákonů, ale spíše z naměřených hodnot.

3 Numerické metody řešení soustav diferenciálních rovnic

Pokud má odporová síla tvar (5) nebo (6), analytické řešení pohybových rovnic by bylo nesnadné. Proto se v programu uchyluji k numerickým metodám. Jejich základní myšlenka je aproximovat rychlost $v(t)$ v dostatečně mnoha časových okamžicích. Aplikace Výstřel z kanónu nabízí výběr ze tří numerických metod:

- Eulerova metoda – nejjednodušší
- Heunova metoda – řadí se mezi tzv. Runge–Kuttovy metody, na něž je možné narazit v praxi

- Adams-Bashforthovy vzorce – pracují na poněkud odlišném principu, využívají informaci ze dvou předchozích časových hladin

Nebudu zacházet do podrobností, ale ještě se zmíním o nedokonalostech numerických výpočtů, které se projevují i v tomto programu.

3.1 Diskretizační a zaokrouhlovací chyby

Nastavíme-li nízký počet snímků za sekundu (a rychlost tím pádem aproximujeme v menším počtu okamžiků), pohyb se jeví trhaný a trajektorie vykazuje „zuby“ – hovoří se pak o *diskretizační chybě* výpočtu.

Mohlo by se zázdat, že co největším zkrácením časového intervalu, po jakém se aktualizuje rychlost, nepřesnost vymizí. Bohužel tím však narůstají chyby jiného druhu, *zaokrouhlovací chyby*. Pramení z přesnosti počítačových výpočtů na konečný počet desetinných míst a toho, že celková chyba střádá i nepřesnosti z minulých časových okamžiků. Dělová koule se pak nepohybuje vyrovnaně, ale osciluje, skáče.

4 Ovládání

Okno aplikace sestává ze dvou sekcí: vykreslovací plochy a ovládacího panelu po pravé straně.

4.1 Vykreslovací plocha

Tažením myši při stisknutém levém tlačítku se otáčí hlaveň děla do patřičného sklonu. Při tom se mění směr počáteční rychlosti, nikoliv však její velikost.

4.2 Ovládací panel

Tlačítkem *Přehrát* se spouští a pozastavuje animace projektilu vyletujícího z děla. Animace se zastavuje tlačítkem *Stop*. Během animace se volitelně vykresluje trajektorie dělové koule. Její průběžná poloha v metrech se vypisuje v horní části ovládacího panelu. Přímo v tomto panelu se nastavuje

- FPS (*frames per second*) – počet snímků animace za sekundu, což je zároveň počet vyhodnocení polohy koule při použití numerického řešení
- vodorovná a svislá složka počáteční rychlosti pohybu koule v $m \cdot s^{-1}$
- tíhové zrychlení v $m \cdot s^{-2}$

- odpor prostředí – přepínač rozhoduje o tom, zda bude odpor prostředí při modelování brán v úvahu
- kreslení trajektorie (ano/ne)
- druhé dělo – ve vykreslovací ploše se objeví druhé dělo

Zbývá dvě tlačítka otevírají dialogová okna s možností podrobnějšího nastavení odporové síly nebo parametrů dělové koule.

4.3 Okno odporu prostředí

Uživatel si může zvolit, jaký odpor prostředí bude v modelu vystupovat. Pro dělovou kouli obtékanou laminárně je zde možnost zadat dynamickou viskozitu látkového prostředí, pro turbulentní proudění jeho hustotu. Pokud se uživatel rozhodne pro odporovou sílu s empirickými parametry, zadává činitel b a exponent α . Kromě toho je zde možnost vybrat si numerickou metodu řešení pohybové rovnice. Tlačítko *Výchozí* vrací nastavení do stavu při spuštění aplikace.

4.4 Okno nastavení dělové koule

Střela kanónu je pro účely modelu popsána svou hmotností, hustotou a poloměrem. Přitom třetí vlastnost je vždy určená na základě dvou zbylých a uživatel si vybírá, které veličiny zadá a která se dopočítá. Při změně poloměru dělové koule se upraví i velikost děla.

5 Programová realizace

Obsluhu programu zajišťuje třída `HlavniOkno` zděděná od `Windows.Forms.Form`. Při načtení vytvoří objekty na vykreslovací ploše a spustí metodu `animacniSmycka()`, která obstarává vykreslování pomocí instance třídy `SpravceAnimace`. Správce animace se může nacházet ve stavech:

- animuje
- pauza
- mechanická rovnováha – po skončení animace přirozenou cestou (např. když dělová koule dopadne na zem)
- neanimuje

Od toho se odvíjí, co se vykresluje. Správce animace drží seznam objektů třídy `PohyblivyObjekt`, které si nesou informaci o souřadnicích, okamžité rychlosti, rozměrech, hmotnosti, reprezentujícím obrázku atd.

Při překreslování pohyblivého objektu na nové pozici opravuje správce animace polohu metodami `aplikujTihoveZrychleni()` a `aplikujOdpor()`, náležitými instancemi fyz. třídy `FyzikalniModel`. Průběh metody `aplikujOdpor()` se velmi liší v závislosti na nastaveném odporu, který patří mezi vlastnosti objektu fyz. Případy nulového odporu a laminárního odporu jsou řešeny explicitním vyjádřením rychlosti objektu v daném čase. Pokud má však odporová síla jiný tvar, konstruuje se rychlosti $v(t_i)$ v časových okamžicích t_i jednou z numerických metod řešení diferenciálních rovnic. To spadá do působnosti třídy `NumerickeMetody`. Objekt fyz. volá metody `reseniX(t)` a `reseniY(t)` instance třídy `NumerickeMetody`. Zmíněné funkce zkontrolují, zda už se řešení pro daný nebo blízký čas nachází v lineárním spojovém seznamu k tomu určenému. Pokud ne, přidají dostatečný počet kroků zvolené numerické metody a seznam prodlouží.

Okno `HlavniOkno` využívá techniku *dvojitého bufferingu* pro větší plynulost animace.

5.1 Výčet tříd a jejich metod

Třídy odvozené od `Windows.Forms.Form` mají navíc ještě metody `InitializeComponent()` a `Dispose()`.

- jmenný prostor `VystrelZKanonu`
 - třída `Program`

```
* static void Main()
```
 - třída `FyzikalniModel`

```
* public void aplikujOdpor(PohyblivyObjekt poh,
    float C, float R, long t)
* public void nastavOdpor(Odpor odpor)
* public void nastavTihoveZrychleni(float g)
* public void nastavHustotuProstredi(float ro)
* public void nastavDynamickouViskozitu(float my)
* public void nastavKrokNumReseni(float deltaT)
* public void nastavEmpOdpSilu(double b, double
    alfa)
```

```

* public void pouzivejNumMetodu(NumerickeMetody.Metoda
    m)
* public float hustotaProstredi()
* public float dynamickaViskozita()
* public float tihoveZrychleni()
* public float pxNaM(float pocetPx)
* public float mNaPx(float pocetM)
- třída NumerickeMetody
* public void nastavKoefficienty(float beta, float
    K, float g, float deltaT)
* public void nastavPocatecniPodminku(float reseni0_x,
    float reseni0_y)
* public void pouzivejMetodu(Metoda m)
* private double f_x(double reseni_x, double reseni_y)
* private double f_y(double reseni_x, double reseni_y)
* private void najdiReseniVSeznamu(ref double
    hledaneReseni, LinkedList<double> seznam, double
    tSec)
* public float reseniX(double tSec)
* public float reseniY(double tSec)
* public void vymazReseni()
- třída SpravceAnimace
* public void nastavFPS(float FPS)
* public void nakresliVychodiStav(Graphics g)
* private void nakresliZemi(Graphics g)
* public bool jeNaZemi(PohyblivyObjekt poh)
* private bool jeKolize(PohyblivyObjekt poh1,
    PohyblivyObjekt poh2)
* public void prepniPauzu()
* public void pridejAnimovanyObjekt(PohyblivyObjekt
    poh)
* public void prekresliVCase(Graphics g, long t)
* public void nastavFyzikalniModel(FyzikalniModel
    fyz)
* public void nastavBitmapu(Bitmap bitmapa)public
    void nastavStav(Stav stav)

```

```

        * public Stav ziskejStav()
- třída PohyblivyObjekt
- public int vyska()
- public int sirka()
- public void pohniSeVCase(long t)
- public Image obrazek()
- public void nastavHmotnost(float hmotnost)
- public bool chceKreslitTrajektorii()
- public void nastavKresleniTrajektorie(bool kreslit)
- public void nastavKinematickeVeliciny(float x, float
  y, int minX, int minY, float v0_x, float v0_y, float
  v_x, float v_y)
- public void nastavPozici(float x, float y)
- public void nastavVychoziPozici(float x_0, float
  y_0)
- public void uvestDoKlidu()
- public void nastavZvetseni(float zvetseni)
- public bool jeVKlidu()
- public bool obsahujeBod(int x, int y)
- public void nastavStredOtaceni(int X, int Y)
- private double eukleidNorma(double a, double b)
- public void otoc(float theta)
- public float ziskejZvetseni()
- třída OdnimatelnyObjekt:PohyblivyObjekt – k pohybli-
  vému objektu přidává navíc vlastnost viditelnosti
- třída HlavniOkno
    * private void animacniSmycka()
    * private void HlavniOkno_Load(object sender, EventArgs
      e)
    * private void tlacitkoPrehrat_Click(object sender,
      EventArgs e)
    * private void tlacitkoStop_Click(object sender,
      EventArgs e)

```

```

* private void HlavniOkno_Activated(object sender,
    EventArgs e)
* private void polickoFPS_Leave(object sender,
    EventArgs e)
* private void ctverecekOdpor_CheckedChanged(object
    sender, EventArgs e)
* private void tlacitkoOdpor_Click(object sender,
    EventArgs e)
* private bool zmenaCiselnehoPolicka(TextBox policko,
    double maxCislo, ref float menenyParametr, float
    koef)
* private void polickoPocRychX_Leave(object sender,
    EventArgs e)
* private void polickoPocRychY_Leave(object sender,
    EventArgs e)
* private void polickoG_Leave(object sender, EventArgs
    e)
* private double eukleidNorma(double a, double
    b)
* private void otocHlaven(ref PohyblivyObjekt hlaven,
    ref PohyblivyObjekt delovaKoule, float uhel,
    double stredX, double stredY)
* private void ctverecekTrajektorie_CheckedChanged(object
    sender, EventArgs e)
* private void ctverecekDruhehoDela_CheckedChanged(object
    sender, EventArgs e)
* private void vykreslovaciPlocha_MouseMove(object
    sender, MouseEventArgs e)
* private void tlacitkoDeloveKoule_Click(object
    sender, EventArgs e)
* private void vykreslovaciPlocha_MouseDown(object
    sender, MouseEventArgs e)
* private void vykreslovaciPlocha_MouseUp(object
    sender, MouseEventArgs e)
* private void HlavniOkno_Resize(object sender,
    EventArgs e)
* private void vykreslovaciPlocha_Paint(object
    sender, PaintEventArgs e)

```


– třída OknoOdporu

```
* private void nastavVychoji()
* private void aktualizujPuvVybraneKolecko()
* private void aktualizujPuvEmpSilu()
* private void koleckoLaminarni_CheckedChanged(object
sender, System.EventArgs e)
* private void koleckoTurbulentni_CheckedChanged(object
sender, System.EventArgs e)
* private void koleckoEmpiricka_CheckedChanged(object
sender, System.EventArgs e)
* public FyzikalniModel.Odpor ziskejNastavenyOdpor()
* public float ziskejB()
* public float ziskejAlfa()
* public float ziskejRo()
* public float ziskejMy()
* private void polickoMy_Leave(object sender, System.EventArgs
e)
* private void polickoRo_Leave(object sender, System.EventArgs
e)
* private void tlacitkoVychoji_Click(object sender,
System.EventArgs e)
* public NumerickeMetody.Metoda ziskejNumMetodu()
* private void OknoOdporu_VisibleChanged(object
sender, System.EventArgs e)
* private void polickoB_Leave(object sender, System.EventArgs
e)
* private void polickoAlfa_Leave(object sender,
System.EventArgs e)
* private void roletkaNumMetoda_SelectedIndexChanged(object
sender, System.EventArgs e)
* private void tlacitkoOK_Click(object sender,
System.EventArgs e)
```

– třída OknoDeloveKoule

```
* private void nastavVychoji()
* public float ziskejR()
* public float ziskejRo()
```

```

* public float ziskejM()
* private void koleckoHmotnost_CheckedChanged(object
sender, EventArgs e)
* private void koleckoHustota_CheckedChanged(object
sender, EventArgs e)
* private void koleckoPrumer_CheckedChanged(object
sender, EventArgs e)
* private void polickoHmotnost_Leave(object sender,
EventArgs e)
* private void tlacitkoVychodi_Click(object sender,
EventArgs e)
* private void tlacitkoOK_Click(object sender,
EventArgs e)
* private void OknoDeloveKoule_VisibleChanged(object
sender, EventArgs e)
* private void polickoHustota_Leave(object sender,
EventArgs e)
* private void polickoPolomer_Leave(object sender,
EventArgs e)

```

6 Příklady uživatelského vstupu

1. $v_{0x} = 1 \text{ m} \cdot \text{s}^{-1}$
 $v_{0y} = 1 \text{ m} \cdot \text{s}^{-1}$
Dělová koule nemá dostatečnou rychlost a pouze vypadne z hlavní.
2. $\mu = 40 \text{ Pa} \cdot \text{s}$
Patrně nerealisticky vysoká dynamická viskozita, ale zase lze dobře pozorovat charakteristický tvar balistické křivky, po níž se koule pohybuje.