Numerické modelování fyzikálních dějů

Jindřich Soukup*, Martin Štrof+, Vojtěch Procházka-
Gymnázium Kladno*, SGaGy Kladno+, Gymnázium Brno-
jindra@matfyz.cz*, marfel@seznam.cz+, v.proch@gmail.com

Abstrakt:

Cílem miniprojektu bylo numericky řešit diferenciální rovnice s fyzikálním pozadím. Hlavním tématem naší práce bylo vymodelovat dráhu letu dělové koule po výstřelu. Zpočátku jsme vytvořili model letu koule bezodporovým prostředím, později jsme simulovali i let koule v prostředí s odporem. Potom jsme se zabývali i modelem letu za silného protivětru. K řešení rovnic jsme používali Eulerovu metodu, samotné modelování se odehrávalo v prostředí Excelu a následně v Mathematice.

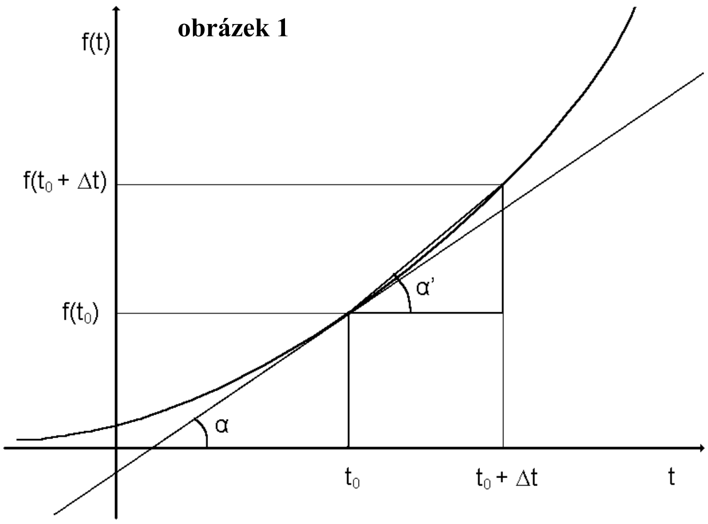
1 Úvod

Pohyb těles v gravitačním poli s odporovým prostředím je jednou z mnoha částí fyziky, kterou je velmi obtížné, ne-li nemožné popsat analytickými metodami. Proto se zde v hojné míře využívá numerických metod, které se nesnaží popsat trajektorii tělesa pomocí diferenciálních rovnic, používaných v analytických metodách, ale využívá postupných výpočtů, které určí výslednou trajektorii. K tomu se dá využít například programů Mathematica a Excel.

V dnešní době, kdy většina obyvatel má přístup k počítači, si takový model může vytvořit každý a názorně si prohlédnout chování těles, která se pohybují kolem něj.

2 Eulerova metoda

V této metodě nahrazujeme diferenciální rovnice, jež neumíme analyticky řešit, rovnicemi diferenčními. Postupně počítáme situaci v různých časech vzdálených od sebe jen malý okamžik a v každém kroku upravíme hodnoty, které se v čase mění. Pokud zvolíme hodně malý časový krok, řešení diferenčních rovnic se od skutečného řešení budou lišit minimálně.



Zde se potýkáme s několika problémy, hlavním z nich jsou derivace. Stačí si ovšem uvědomit, že derivace je $\lim_{\Delta t \to 0} \frac{f(t_0 + \Delta t) - f(t_0)}{\Delta t}$. Pokud tedy zvolíme *Δt* dostatečně malé, derivace budou téměř odpovídat. Situaci vidíme na *obrázku 1*, rovná čára značí tečnu funkce v daném bodě, derivace odpovídá *tgα*, hodnota derivace získaná Eulerovou metodou je *tgα'*.

Ve výpočtech navíc můžeme *Δt* převést na druhou stranu rovnice, protože známe jeho hodnotu.

3 Implementace do tabulkového editoru

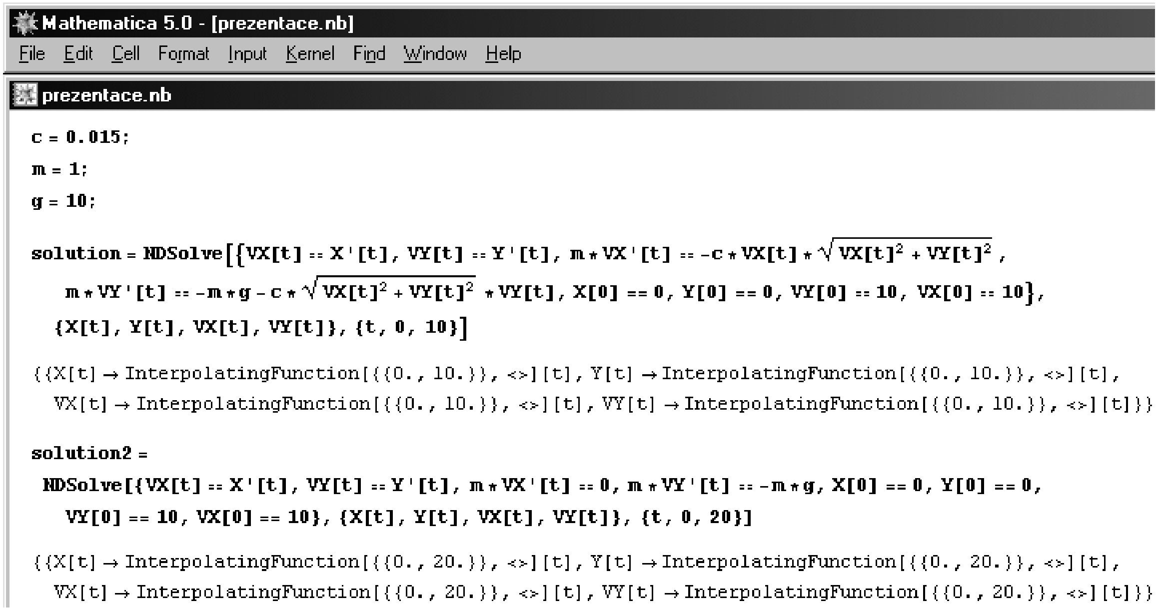
Nyní je čas přejít od teorie k praxi. Všechny dřívější poznatky nyní použijeme při simulování balistické dráhy dělové koule v prostředí s odporem vzduchu.

t	x	y	v_x	v_y
0	0	0	35,35534	35,35534
	$x_{n+1} = x_n + v_x \cdot \Delta t$	$y_{n+1} = y_n + v_y \cdot \Delta t$	$vx_{n+1} = vx_n - \frac{k \cdot v \cdot vx}{m} \cdot \Delta t$	$vy_{n+1} = vy_n - g \cdot \Delta t - \frac{k \cdot v \cdot vy}{m} \cdot \Delta t$
0,001	0,035355	0,035355	35,17856	35,16875
0,002	0,070534	0,070524	35,00357	34,9840
0,003	0,105537	0,105508	34,83035	34,80106
0,004	0,140368	0,140309	34,65885	34,6199
0,005	0,175027	0,174929	34,48907	34,44050

V první řádce jsou sledované veličiny, jako rychlost „Vx“ v x-ovém směru a rychlost „Vy“ v y-ovém směru. X a Y jsou souřadnice pro graf:

4 Modelování v Mathematice

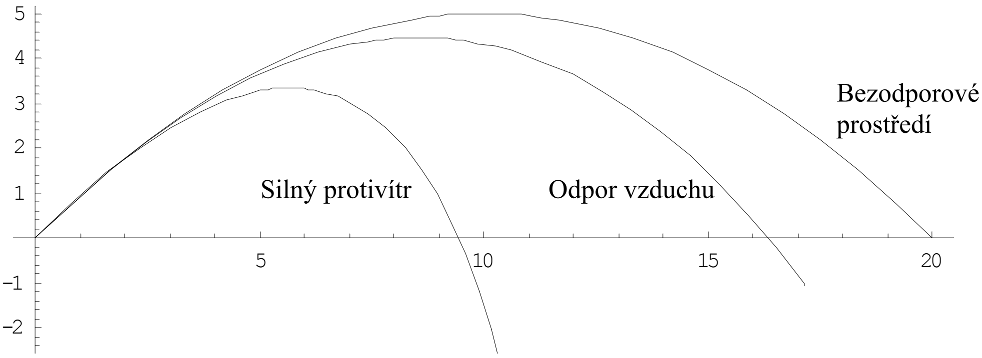
Program Mathematica umí řešit diferenciální rovnice automaticky, vybere vždy nejvhodnější metodu a výsledek zobrazí jako graf, jen vědět, jak na to. K numerickému řešení diferenciálních rovnic slouží příkaz NDSolve. Ukázku takovéhoto zadání si můžete prohlédnout na *obrázku 2*.


Mathematica 5.0 - [prezentace.nb]
File Edit Cell Format Input Kernel Find Window Help
prezentace.nb

```
c = 0.015;
m = 1;
g = 10;

solution = NDSolve[{VX[t] == X'[t], VY[t] == Y'[t], m*VX'[t] == -c*VX[t]*Sqrt[VX[t]^2 + VY[t]^2],
    m*VY'[t] == -m*g - c*Sqrt[VX[t]^2 + VY[t]^2]*VY[t], X[0] == 0, Y[0] == 0, VY[0] == 10, VX[0] == 10},
   {X[t], Y[t], VX[t], VY[t]}, {t, 0, 10}]

{{X[t] -> InterpolatingFunction[{{0., 10.}}, <>][t], Y[t] -> InterpolatingFunction[{{0., 10.}}, <>][t],
  VX[t] -> InterpolatingFunction[{{0., 10.}}, <>][t], VY[t] -> InterpolatingFunction[{{0., 10.}}, <>][t]}}

solution2 =
 NDSolve[{VX[t] == X'[t], VY[t] == Y'[t], m*VX'[t] == 0, m*VY'[t] == -m*g, X[0] == 0, Y[0] == 0,
   VY[0] == 10, VX[0] == 10}, {X[t], Y[t], VX[t], VY[t]}, {t, 0, 20}]

{{X[t] -> InterpolatingFunction[{{0., 20.}}, <>][t], Y[t] -> InterpolatingFunction[{{0., 20.}}, <>][t],
  VX[t] -> InterpolatingFunction[{{0., 20.}}, <>][t], VY[t] -> InterpolatingFunction[{{0., 20.}}, <>][t]}}
```

obrázek 2

Na obrázku 3 je graf vygenerovaný programem Mathematica, na kterém je znázorněn pohyb dělové koule při různých okolních podmínkách.



obrázek 3

5 Shrnutí

V Excelu, ale především v programu Mathematica, který používá speciálních metod při řešení numerických rovnic, jsme dostali výsledky, které s hodně velkou přesností odpovídají skutečnosti. Avšak při extrémních situacích bychom mohli zjistit vzrůstající odchylky. Je to způsobeno krokováním, kterého zde využíváme. Ve skutečnosti se tělesa pohubují absolutně plynule a sebejemnější krokování vždy vede k nepřesnostem.

Poděkování

Děkujeme Fakultě jaderné a fyzikálně inženýrské a všem organizátorům Fyzikálního týdne za umožnění práce na miniprojektu. Poděkování rovněž posíláme i našim supervisorům, a to především za odborné konzultace.

Reference:

[1] R. FEYNMAN *Přednášky z fyziky I*
[2] SPN *Matematické, fyzikální a chemické tabulky*
[3] T.BEDNÁRIK, M.FRANĚK, J.NÁVRAT A R. SMRŽ
Numerické modelování pohybů v gravitačním poli