VYUŽITÍ MATLABU JAKO MOTIVAČNÍHO PROSTŘEDKU VE VÝUCE FYZIKY NA STŘEDNÍCH ŠKOLÁCH

J. Tesař, P. Bartoš

Jihočeská univerzita, Pedagogická fakulta, Katedra fyziky,

Jeronýmova 10, 371 15 České Budějovice

Abstrakt

V příspěvku se zabýváme možností využití programového balíku MATLAB jako motivačního prostředku ve výuce fyziky na středních školách. Prezentujeme jednoduchou ukázku modelu difúze částic neutrálního plynu, na které zároveň vysvětlujeme základní obraty počítačové fyziky používané při deterministickém přístupu částicového modelování – algoritmus pro pohyb částic, srážkové procesy mezi molekulami, jevy na okraji pracovní oblasti atp.

1 Úvod

Počítačové modelování si nalezlo pevné místo prakticky ve všech odvětvích lidské činnosti a s tím je spojeno i pronikání tohoto relativně mladého vědního oboru také do našich škol. Na vysokých školách technického či přírodovědného zaměření je nabídka základního kurzu počítačového modelování již prakticky standardem. Čím dál častěji se také setkáváme s využitím počítačových simulací učiteli středních škol – ne snad za účelem detailního studia daného jevu, ale spíše ve snaze vytvořit u studentů názornou a atraktivní cestou trvalý poznatek.

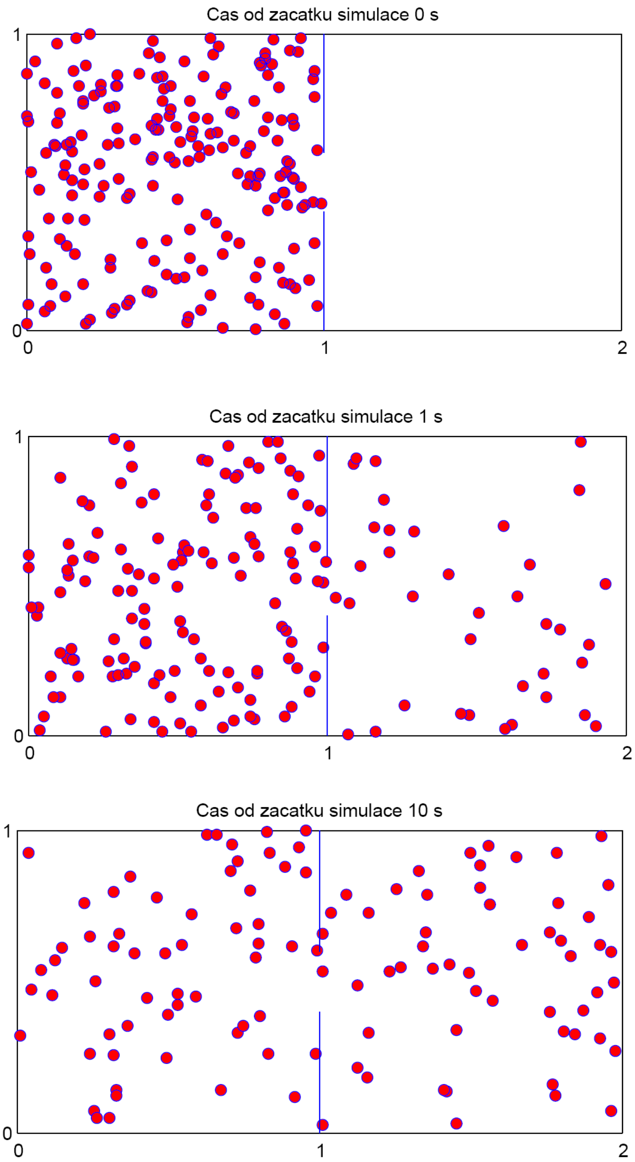
V následujícím krátkém příspěvku se pokusíme demonstrovat možnosti využití programovacího prostředí MATLAB pro tento účel. Je zřejmé, že tento program není jediným možným prostředkem, jak dosáhnout vytvoření působivé a přitom reálné simulace, nicméně jednoduchá a srozumitelná syntaxe tohoto jazyka s řadou předpřipravených funkcí neodvádí od podstaty studovaného jevu. K vytvoření modelu navíc postačí zpravidla znalost pouze několika základních funkcí. Získané zkušenosti s programem ve škole mohou studenti využít ve své praxi, neboť MATLAB je využíván na celé řadě odborných pracovišť.

Počítačové modely se zpravidla dělí do dvou základních skupin podle toho, zda je systém popisován jako kontinuum nebo zda na něj nahlížíme na úrovni jednotlivých částic, které toto kontinuum tvoří. V prvém případě mluvíme o spojitém modelování, při kterém řešíme zpravidla soustavu několika mála diferenciálních rovnic (k tomu lze například využít programového prostředí COMSOL Multiphysics). Druhý přístup bývá nazýván částicové modelování a zde je nutno hledat trajektorii obvykle velkého množství částic pohybujících se podle základních pravidel dynamiky. Každá z těchto trajektorií je popsána soustavou diferenciálních rovnic, takže částicové metody modelování jsou mnohem více náročné na spotřebu strojového času. Jejich výhodou však je, že umožňují detailnější popis studovaného jevu, což většinou umožňuje získat výsledky s vyšší vypovídající hodnotou.

Částicové techniky jsou dále děleny na techniky stochastické, kdy je využíváno počtu pravděpodobnosti, a techniky deterministické. Tyto techniky bývají často vzájemně kombinovány (zpravidla v případě nedostatku vstupních dat).

Zaměřme se v tomto příspěvku na některé základní obraty deterministické modelovací metody na příkladu difúze neutrálních částic. Předpokládejme, že máme pracovní oblast rozdělenou na dvě stejně velké oblasti pro částice nepropustnou překážkou. V obou oblastech máme plyn o stejné koncentraci (ten nám představuje pozadí), v levé polovině této oblasti pak umístíme navíc N částic plynu, jehož difúzi do pravé části pracovní oblasti budeme sledovat. Pro jednoduchost předpokládejme, že všechny tyto částice můžeme považovat za body zanedbatelné hmotnosti a že všechny částice mají v průběhu experimentu stejnou rychlost. Dále předpokládejme, že problém řešíme jako dvoudimenzionální případ (pro lepší vizualizaci a pochopení studenty, převedení do 3D prostoru je zřejmé).

V čase t_0 odstraníme část překážky tak, jak je to patrné z obrázku 1 nahoře. Spustíme výpočet a sledujeme, jak částice pronikají z levé strany pracovní oblasti do pravé, až se počty částic na obou stranách přibližně vyrovnají (obrázek 1 dole). Změnou vstupních parametrů (střední volná dráha částic mezi srážkami, jejich rychlost, velikost štěrbiny atp.) tak můžeme se studenty simulovat vlastnosti jevu nazývaného difúze.



Obrázek 1: Rozložení částic v pracovní oblasti v čase 0 s, 1 s a 10 s od počátku výpočtu.

2 Popis počítačového modelu

Zdrojový kód tohoto modelu bude sestávat z několika základních částí.

1. Část, v níž zadáme všechny potřebné parametry modelu a dopočítání ostatních fyzikálních veličin (řádky 1-13 zdrojového kódu – viz dále).
2. Rozehrání počátečních poloh, rychlostí a náhodných volných drah jednotlivých částic (řádky 14-22 zdrojového kódu).
3. Eulerův algoritmus (řádky 59-65 zdrojového kódu).

4. Ošetření událostí na okraji pracovní oblasti a na překážce (řádky 85-103 zdrojového kódu).
5. Zavedení srážkových procesů v plynu (řádky 18 a 119-130 zdrojového kódu)
6. Grafický výstup provedené simulace (řádky 24-32 a 49-55 zdrojového kódu).

Hlavní funkce modelu je označena jako *main*, ostatní funkce slouží pro provedení dílčích výpočtů a jsou volány uvnitř hlavní funkce. Zdrojový kód celého programu je v následující kapitole.

2.1 Zadání počátečních parametrů a dopočítání ostatních veličin

V běžném plynu bývá koncentrace částic vysoká a rychlosti částic dosahují poměrně velkých hodnot. Pro potřeby takové simulace by bylo potřeba volit pracovní oblast velmi malých rozměrů a případně využít výkonný počítač. Jelikož právě tento požadavek nebývá na většině škol splněn a provedení simulace s reálnými parametry tudíž nemusí přinést očekávaný efekt, je výhodnější, když počáteční parametry zvolíme „nefyzikálně“ tak, aby byly dobře pozorovatelné podstatné jevy studovaného jevu. V našem případě jsme zvolili velikost hrany pracovní oblasti 1 m, rychlost částic 5 $m.s^{-1}$, střední volnou dráhu částic mezi srážkami 0,5 m a počet sledovaných částic 200. Studenty na tento fakt upozorníme a výklad můžeme doplnit výpočtem reálných hodnot takto zvolených veličin.

2.2 Rozehrání počátečních poloh, rychlostí a náhodných volných drah jednotlivých částic

Počáteční polohy částic jsou určeny nagenerováním bodu ve čtverci pomocí funkce *rand*. Směr rychlosti v polárních souřadnicích můžeme zadat nagenerováním náhodného úhlu φ nebo lze také využít postupu, kdy ve čtverci se středem v bodě [0, 0] a hranami o délce 2 nagenerujeme bod, otestujeme, zda se tento bod nachází uvnitř kružnice o poloměru 1 a posléze normujeme souřadnice x a y vzdáleností tohoto bodu od středu *r*. Takto získané složky poté využijeme k rozehrání náhodného směru vektoru rychlosti.

Předpis pro rozehrávání náhodné volné dráhy získáme ze vzorce pro pravděpodobnost Poissonova jevu a lze psát

$$\lambda_{\mathrm{i}} = -\lambda_{\mathrm{str}} \ln \gamma, \tag{1}$$

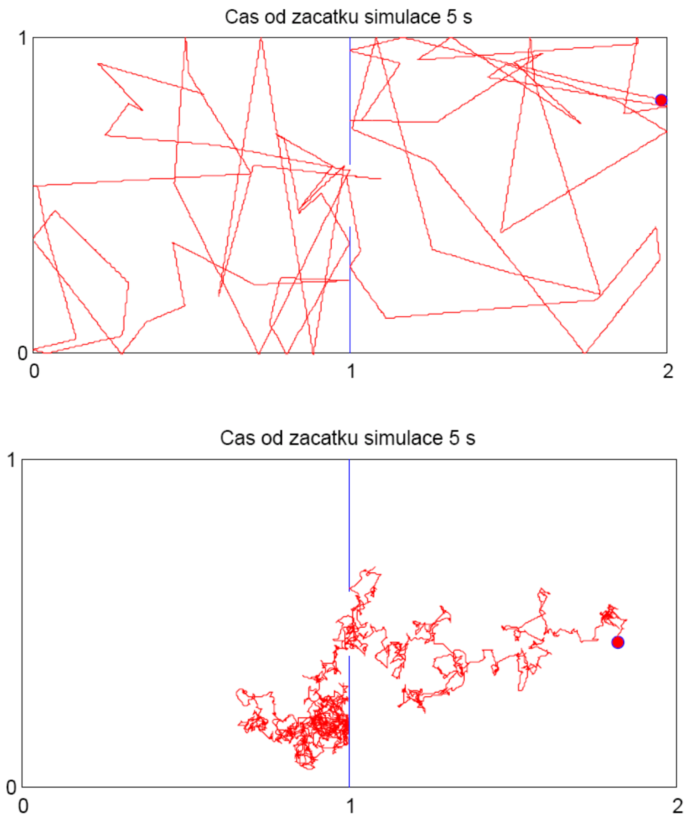
kde λ_{i} je střední volná dráha i-té částice, λ_{str} je střední volná dráha daného typu částic a γ je náhodné číslo s rovnoměrným rozdělením na intervalu $(0,1)$.

2.3 Pohyb částic

Pohyb jednotlivých částic uvnitř pracovní oblasti lze realizovat několika postupy numerické matematiky. Nejjednodušším postupem je tzv. Eulerův algoritmus, který diskretizuje rovnici pro polohu a rychlost částice takto:

$$\begin{aligned} \vec{r} &= \vec{r}_0 + \vec{v}t + \frac{\vec{a}t^2}{2} \\ \vec{v} &= \vec{v}_0 + \vec{a}t \\ \vec{F} &= \ldots \end{aligned} \quad \Rightarrow \quad \begin{aligned} \vec{r}^{\,k+1} &= \vec{r}^{\,k} + \vec{v}^{\,k}\Delta t + \frac{1}{2m}\vec{F}^{\,k}\Delta t^2 \\ \vec{v}^{\,k+1} &= \vec{v}^{\,k} + \frac{\vec{F}^{\,k}}{m}\Delta t \\ \vec{F}^{\,k+1} &= \ldots \end{aligned}$$

Pohyb každé částice je nutné sledovat zvlášť s daným časovým krokem Δt.



Obrázek 2: Trajektorie jedné částice v případě střední volné dráhy 0,5 metru (horní obrázek) a 0,01 metru (dolní obrázek).

2.4 Pohyb částic

Dále je v modelu nutné ošetřit chování částice na okraji pracovní oblasti. V našem případě předpokládáme, že okraj pracovní oblasti představuje pro částici nepřekonatelnou překážku (stěnu, od které se částice odrazí). V tomto případě tedy postačí pouze vyhledat ty částice, které v daném časovém kroku opustily pracovní oblast, a změnit znaménko složky rychlosti, která je kolmá ke stěně na opačné. Při vhodně zvoleném časovém kroku lze zanedbat vzdálenost, po kterou se částice pohybovala mimo pracovní oblast.

2.5 Srážkové procesy

V plynu dochází k neustálým srážkám mezi částicemi. V našem modelu realizujeme srážky tak, že na počátku výpočtu přidělíme každé částici náhodnou volnou dráhu, od níž v průběhu výpočtu postupně odečítáme uraženou vzdálenost. V okamžiku, kdy daná částice urazí celou náhodnou volnou dráhu, realizujeme srážku, která spočívá v nagenerování nového náhodného směru, a zároveň částici přidělíme novou hodnotu náhodné volné dráhy.

Právě tento parametr charakterizuje rychlost šíření částic do volného prostoru. Program MATLAB umožňuje jednoduše zvýraznit trajektorii jedné částice (viz. obrázek 2). Provedeme-li simulaci pro různé parametry, umožníme vytvoření názorné představy o faktorech ovlivňujících rychlost difúze částic.

2.6 Grafický výstup

Program MATLAB je velmi dobře známý svým kvalitním grafickým výstupem. Oproti jiným programovacím jazykům umožňuje vytvoření obrázku zadáním jednoho příkazu. Jelikož náš model je nastaven tak, aby průběžně vykresloval pohyb částic v pracovní oblasti, je nejprve vykreslen graf z počátečních hodnot polohy a následně jsou tyto hodnoty aktualizovány příkazem *set*. Překreslení obrázku je vynuceno příkazem *drawnow*.

3 Výpis zdrojového kódu

```
function main()
global delta_t L_x L_y stredni_volna_draha v

% Definice konstant a parametru
L_x=1;       % velikost pracovni oblasti ve smeru osy x, v niz generujeme castice
L_y=1;       % totez - smer y

v=5;         % pocatecni rychlost castic
N=200;       % pocet nagenerovanych castic

delta_t=1e-3;               % casovy interval pouzity v Eulerove algortmu
stredni_volna_draha=0.5;    % stredni volna draha castic

% Rozehrani poloh, stredni volne drahy a smeru rychlosti jednotlivych castic
x=(L_x)*rand(1, N);
y=(L_y)*rand(1, N);

lambda=-stredni_volna_draha*log(rand(1, N));

[smer_x, smer_y]=generuj_smer_vektoru(N);
v_x=v*smer_x;
v_y=v*smer_y;

% Graficky vystup
h=plot(x,y, 'o', 'markerfacecolor', 'r');
set(gca, 'XTick', [0, L_x, 2*L_x], 'YTick', [0, L_y])
axis equal
axis([0, 2*L_x, 0, L_y])
set(gcf, 'color','w')
line([1 1],[0 0.4])
line([1 1],[0.6 1])
title('Cas od zacatku simulace 0 s');

% Cyklus resici pohyb castic
for i=1:1:1000
    x_stara=x;      % ulozeni soucasnych souradnic polohy
    y_stara=y;

    [x, y]=euler(x, y, v_x, v_y);    % provedeni Eulerova algoritmu
    posun=sqrt((x_stara-x).^2+(y_stara-y).^2);  % zmenseni n. v. drahy
    lambda=lambda-posun;

    % osetri, co se stane s casticI na okraji pracovni oblasti
    [x, y, v_x, v_y]=okraj_oblasti(x, x_stara, y, v_x, v_y);

    % osetri srazkove procesy mezi pozadim a casticemi
    [v_x, v_y, lambda]=srazka(v_x, v_y, lambda);

    % graficky zpracuje data
    set(h, 'xdata', x, 'ydata', y);
    titulek=delta_t*i;
    titulek=num2str(titulek);
    titulek=['Cas od zacatku simulace ' titulek ' s'];
    title(titulek)
    drawnow
end      % konec cyklu for
```

```
% ***************************************************
function [x, y]=euler(x, y, v_x, v_y)
% Pohyb castic v pracovni oblasti
global delta_t

x=x+v_x*delta_t;
y=y+v_y*delta_t;


% ***************************************************
function [smer_x, smer_y]=generuj_smer_vektoru(pocet)
% Tato funkce rozehrava smer vektroru
smer_x=[];
smer_y=[];
while length(smer_x)<pocet
    smer_x_pom=2*rand(1,1)-1;
    smer_y_pom=2*rand(1,1)-1;
    r=sqrt(smer_x_pom.^2+smer_y_pom.^2);

    if r<=1
        smer_x=[smer_x smer_x_pom/r];
        smer_y=[smer_y smer_y_pom/r];
    end
end


% ***************************************************
function [x, y, v_x, v_y]=okraj_oblasti(x, x_old, y, v_x, v_y)
global L_x L_y
% Funkce osetri udalost na okraji pracovni oblasti

% Leva a prava strana - vrati castici zpet
index=find(x<=0 | x>=2*L_x);
v_x(index)=-v_x(index);

% Membrana - bud castici propusti nebo zajisti odraz od prekazky
index=find(((x>=L_x) & (x_old<=L_x))...
    |((x<=L_x) & (x_old>=L_x)));
index2=find((y(index)<=0.4) | (y(index)>=0.6));
v_x(index(index2))=-v_x(index(index2));
x(index(index2))=L_x-(x(index(index2))-L_x);

% Horni a spodni okraj - vraci castici zpet
index=find(y<=0 | y>=L_y);
v_y(index)=-v_y(index);
```

```
% ***************************************************
function [v_x, v_y, lambda]=srazka(v_x, v_y, lambda)
global stredni_volna_draha v
% Funkce rozhrava novou nahodnou volnou drahu v pripadech, kdy nastala
% srazka castice  neutralnim pozadim.

index=find(lambda<=0);

if isempty(index)
else
    lambda(index)=-stredni_volna_draha*log(rand(1,length(index)));

```

4 Závěr

Prezentovaný model lze samozřejmě i dále rozšiřovat. Lze například uvažovat, že rozdělení rychlostí molekul není dáno pouze jednou hodnotou, ale že rozdělení rychlostí je maxwellovské. Pokud chceme zcela potlačit programátorskou stránku do pozadí a používat program jako „blackbox“, můžeme k modelu vytvořit efektní grafické rozhraní pomocí Graphical User Interface, v němž bude možné měnit požadované parametry.

Vytvořený model umožňuje dle našeho názoru lepší pochopení základních pojmů molekulové fyziky a jejich vzájemných souvislostí – např. rozdíl mezi pojmy náhodná volná dráha a střední volná dráha částic. Vytvoří názornou a déletrvající představu o fyzikálním jevu zvaném difúze a v kombinaci s dalšími prostředky, jako je například fyzikální experiment, dokáže motivovat studenty k většímu zájmu o přírodní vědy na vysoké škole. Kromě toho, seznámení se s MATLABem a základními obraty numerické matematiky již v průběhu studia na SŠ může být pro studenty cennou zkušeností pro jejich následnou praxi v zaměstnání. Za vhodné považujeme jeho zařazení do fyzikálního semináře pro žáky s hlubším zájmem o fyziku a programování.

Jiří Tesař
Jihočeská univerzita v Českých Budějovicích
Pedagogická fakulta, Katedra fyziky
Jeronýmova 10
371 15 České Budějovice
e-mail: raset@pf.jcu.cz

Petr Bartoš
Jihočeská univerzita v Českých Budějovicích
Pedagogická fakulta, Katedra fyziky
Jeronýmova 10
371 15 České Budějovice
e-mail: bartos-petr@seznam.cz