

Příklady do Vybraných kapitol z matematické fyziky

Petr Nitsche

OFMF

petr-nitsche@seznam.cz

1 Rovnice „mělké vody“

V angličtině Shallow water equations (zkráceně SWEs) je sada hyperbolických parciálních diferenciálních rovnic, které se používají k modelování vln zejména v případě kdy vlnová délka je větší než hloubka vody. SWEs jsou odvozeny z Navier-Stokesovy rovnice (v odvození se předpokládá, že vertikální rychlost je dostatečně malá aby se dala zanedbat).

Sada parciálních diferenciálních rovnic tvořících SWEs je následující:

$$\begin{aligned}\frac{\partial h}{\partial t} + \frac{\partial(hu)}{\partial x} + \frac{\partial(hv)}{\partial y} &= 0 \\ \frac{\partial(hu)}{\partial t} + \frac{\partial}{\partial x}\left(hu^2 + \frac{1}{2}gh^2\right) + \frac{\partial(huv)}{\partial y} &= 0 \\ \frac{\partial(hv)}{\partial t} + \frac{\partial(huv)}{\partial x} + \frac{\partial}{\partial y}\left(hv^2 + \frac{1}{2}gh^2\right) &= 0\end{aligned}$$

kde h je celková výška sloupce tekutiny, g je gravitační zrychlení, u a v označují rychlosti ve směru x a y . Parciální derivace seskupíme do vektorů a přepíšeme jako jednu hyperbolickou parciální diferenciální rovnici.

Vektory:

$$\begin{aligned}H &= \begin{pmatrix} h \\ hu \\ hv \end{pmatrix} \\ U(H) &= \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{pmatrix} \\ V(H) &= \begin{pmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{pmatrix}\end{aligned}$$

Hyperbolická PDR:

$$\frac{\partial H}{\partial t} + \frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} = 0$$

kde H je dvourozměrná matice výškového posunutí. U je dvoudimenzionální matice, která ukládá rychlost každého bodu ve směru x (kladné číslo, představuje rychlost v kladném

směru osy x a naopak). V je dvoudimenzionální matice, která ukládá rychlost každého bodu ve směru y .

K vyřešení rovnice použijeme numerickou metodu známou jako Lax-Wendroffova metoda pro řešení hyperbolických parciálních diferenciálních rovnic, na základě konečných rozdílů. Na rozdíl od Eulerovi metody, která vypočítává každý krok funkce, metoda Lax-Wendroffova nejdříve vypočítá půl kroku (first half step) a pak pomocí výsledku z poloviny kroku vypočítá celý krok (second half step). To je formálně zapsáno následovně.

Předpokládejme, že máme takto formulovanou rovnici:

$$\frac{\partial f(x, t)}{\partial t} = \frac{\partial g(f(x, t))}{\partial x}$$

První krok je:

$$\frac{f_{i+1/2}^{n+1/2} - \frac{f_i^n + f_{i+1}^n}{2}}{(1/2) * \Delta t} = \frac{g_{i+1}^n - g_i^n}{\Delta x}$$

Druhý krok je:

$$\frac{f_i^{n+1} - f_i^n}{\Delta t} = \frac{g_{i+1/2}^{n+1/2} - g_{i-1/2}^{n+1/2}}{\Delta x}$$

Snadno ukážeme, že toto řešení platí i pro SWEs. Za f dosadíme H a za $g(f)$ dosadíme $U(H)$.

Rovnici potom můžeme přepsat jako:

$$Hx_{i+1/2}^{n+1/2} = \frac{H_{i+1}^n + H_i^n}{2} + (2 * \Delta t) \frac{U_{i+1}^n - U_i^n}{\Delta x}$$

$$Hy_{j+1/2}^{n+1/2} = \frac{H_{j+1}^n + H_j^n}{2} + (2 * \Delta t) \frac{V_{j+1}^n - V_j^n}{\Delta y}$$

Protože H je dvourozměrné, musí se krok spočítat dvakrát, jednou pro každý rozměr. Půl kroky pro U a V jsou vypočteny stejným způsobem a uloženy jako Ux , Uy , Vx a Vy . Konečná rovnice pro H může být napsána jako:

$$H_{i,j}^{n+1} = H_{i,j}^n + \Delta t \frac{Ux_{i+1/2}^{n+1/2} - Ux_{i-1/2}^{n+1/2}}{\Delta x} + \Delta t \frac{Vy_{i+1/2}^{n+1/2} - Vy_{i-1/2}^{n+1/2}}{\Delta y}$$

Předchozí rovnice pro H napíšeme do programu MATLAB následovně:

$$Hx(i, j) = (H(i + 1, j + 1) + H(i, j + 1))/2 - dt/(2 * dx) * (U(i + 1, j + 1) - U(i, j + 1));$$

$$Hy(i,j) = (H(i+1,j+1) + H(i+1,j))/2 - dt/(2 * dx) \\ * (V(i+1,j+1) - V(i+1,j));$$

$$H(i,j) = H(i,j) - (dt/dx) * (Ux(i,j-1) - Ux(i-1,j-1)) - (dt/dy) \\ * (Vy(i-1,j) - Vy(i-1,j-1));$$

Zdrojový kód výpočtu v programu MATLAB

```
clf;
```

```
clear all;
```

```
n = 50;          % velikost sítě
```

```
dt = 0.01;       % integrační krok
```

```
dx = 1;
```

```
dy = 1;
```

```
g = 9.8;
```

```
H = ones(n+2,n+2);    % matice pro výšku
```

```
U = zeros(n+2,n+2);    % x rychlost
```

```
V = zeros(n+2,n+2);    % y rychlost
```

```
% vykreslení sítě
```

```
grid = surf(H);
```

```
axis([1 n 1 n 0 3]);
```

```
title('Vlny ve vaně');
```

```
xlabel('Vana');
```

```
ylabel('Vana');
```

```
zlabel('Výška vlny');
```

```
hold all;
```

```
% počáteční vychýlení
```

```
[x,y] = meshgrid( linspace(-3,3,10) );
```

```
R = sqrt(x.^2 + y.^2) + eps;
```

```
Z = (sin(R)./R);
```

```
Z = max(Z,0);
```

```
% přidat vychýlení do matice pro výšku
```

```
w = size(Z,1);
```

```
i = 10:w+9; j = 20:w+19;
```

```
H(i,j) = H(i,j) + Z;
```

```
% prázdné matice pro half step výpočty
```

```
Hx = zeros(n+1,n+1); Hy = zeros(n+1,n+1);
```

```
Ux = zeros(n+1,n+1); Uy = zeros(n+1,n+1);
```

```
Vx = zeros(n+1,n+1); Vy = zeros(n+1,n+1);
```

```
while 1==1
```

```
% překreslí síť
```

```
set(grid,'zdata', H);
```

```
drawnow
```

```
% Okrajové podmínky pro odraz
```

```
H(:,1) = H(:,2); U(:,1) = U(:,2); V(:,1) = -V(:,2);
```

```
H(:,n+2) = H(:,n+1); U(:,n+2) = U(:,n+1); V(:,n+2) = -V(:,n+1);
```

```
H(1,:) = H(2,:); U(1,:) = -U(2,:); V(1,:) = V(2,:);
```

```
H(n+2,:) = H(n+1,:); U(n+2,:) = -U(n+1,:); V(n+2,:) = V(n+1,:);
```

```
% First half step
```

```
i = 1:n+1;
```

```
j = 1:n+1;
```

```
% výška
```

```
Hx(i,j) = (H(i+1,j+1)+H(i,j+1))/2-dt/(2*dx)*(U(i+1,j+1)-U(i,j+1));
```

```
Hy(i,j) = (H(i+1,j+1)+H(i+1,j))/2-dt/(2*dy)*(V(i+1,j+1)-V(i+1,j));
```

% x hybnost

$$U_x(i,j) = (U(i+1,j+1)+U(i,j+1))/2-dt/(2*dx)*(U(i+1,j+1).^2./H(i+1,j+1)-U(i,j+1).^2./H(i,j+1) + ...$$

$$g/2*H(i+1,j+1).^2-g/2*H(i,j+1).^2);$$

$$U_y(i,j) = (U(i+1,j+1)+U(i+1,j))/2-dt/(2*dy)*((V(i+1,j+1).*U(i+1,j+1)./H(i+1,j+1)) - ...$$

$$(V(i+1,j).*U(i+1,j)./H(i+1,j))));$$

% y hybnost

$$V_x(i,j) = (V(i+1,j+1)+V(i,j+1))/2-dt/(2*dx)*((U(i+1,j+1).*V(i+1,j+1)./H(i+1,j+1))-(U(i,j+1).*V(i,j+1)./H(i,j+1)));$$

$$V_y(i,j) = (V(i+1,j+1)+V(i+1,j))/2-dt/(2*dy)*((V(i+1,j+1).^2./H(i+1,j+1)+g/2*H(i+1,j+1).^2)-(V(i+1,j).^2./H(i+1,j) + ...$$

$$g/2*H(i+1,j).^2));$$

% Second half step

i = 2:n+1;

j = 2:n+1;

% výška

$$H(i,j) = H(i,j)-(dt/dx)*(U_x(i,j-1)-U_x(i-1,j-1))-(dt/dy)*(V_y(i-1,j)-V_y(i-1,j-1));$$

% x hybnost

$$U(i,j) = U(i,j)-(dt/dx)*((U_x(i,j-1).^2./H_x(i,j-1) + g/2*H_x(i,j-1).^2)-(U_x(i-1,j-1).^2./H_x(i-1,j-1) + ...$$

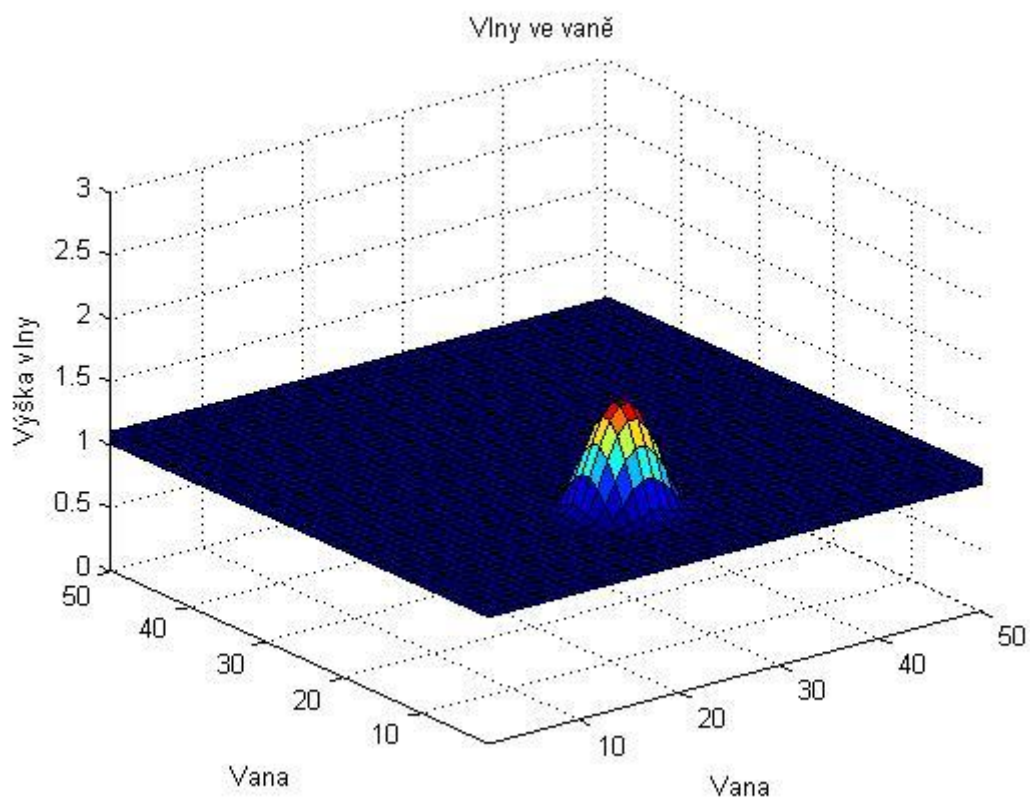
$$g/2*H_x(i-1,j-1).^2))-(dt/dy)*((V_y(i-1,j).*U_y(i-1,j)./H_y(i-1,j))-(V_y(i-1,j-1).*U_y(i-1,j-1)./H_y(i-1,j-1))));$$

% y hybnost

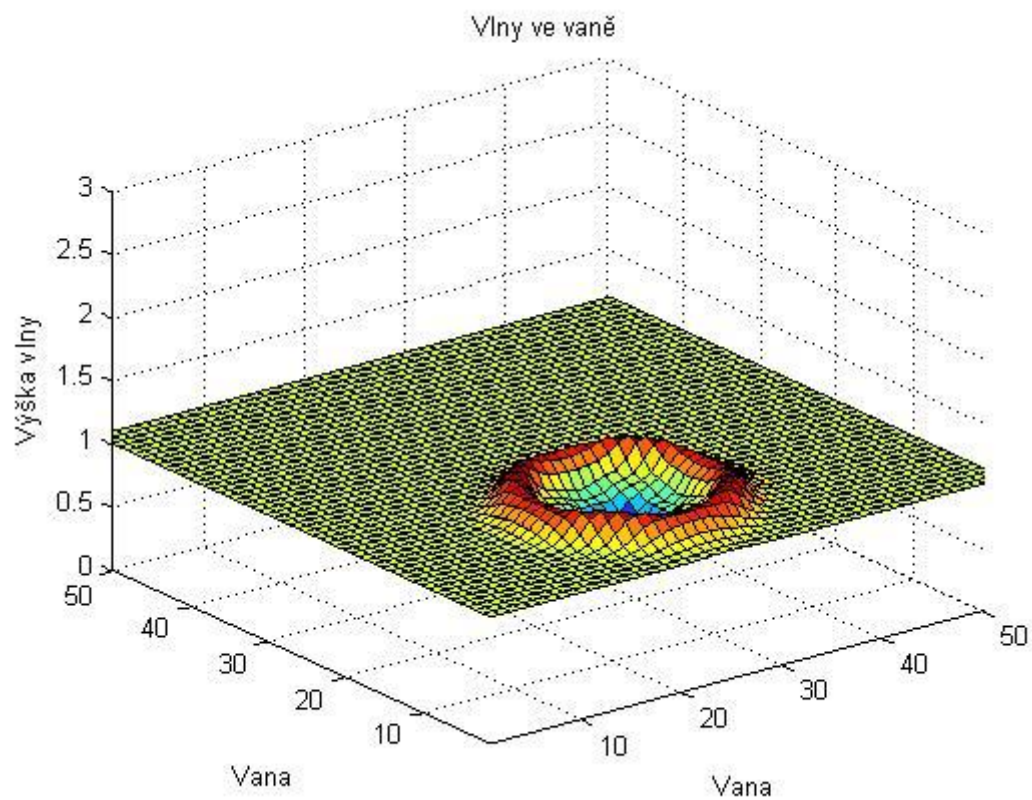
$$V(i,j) = V(i,j)-(dt/dx)*((U_x(i,j-1).*V_x(i,j-1)./H_x(i,j-1))-(U_x(i-1,j-1).*V_x(i-1,j-1)./H_x(i-1,j-1)))-(dt/dy) * ...$$

$$((V_y(i-1,j).^2./H_y(i-1,j) + g/2*H_y(i-1,j).^2)-(V_y(i-1,j-1).^2./H_y(i-1,j-1) + g/2*H_y(i-1,j-1).^2));$$

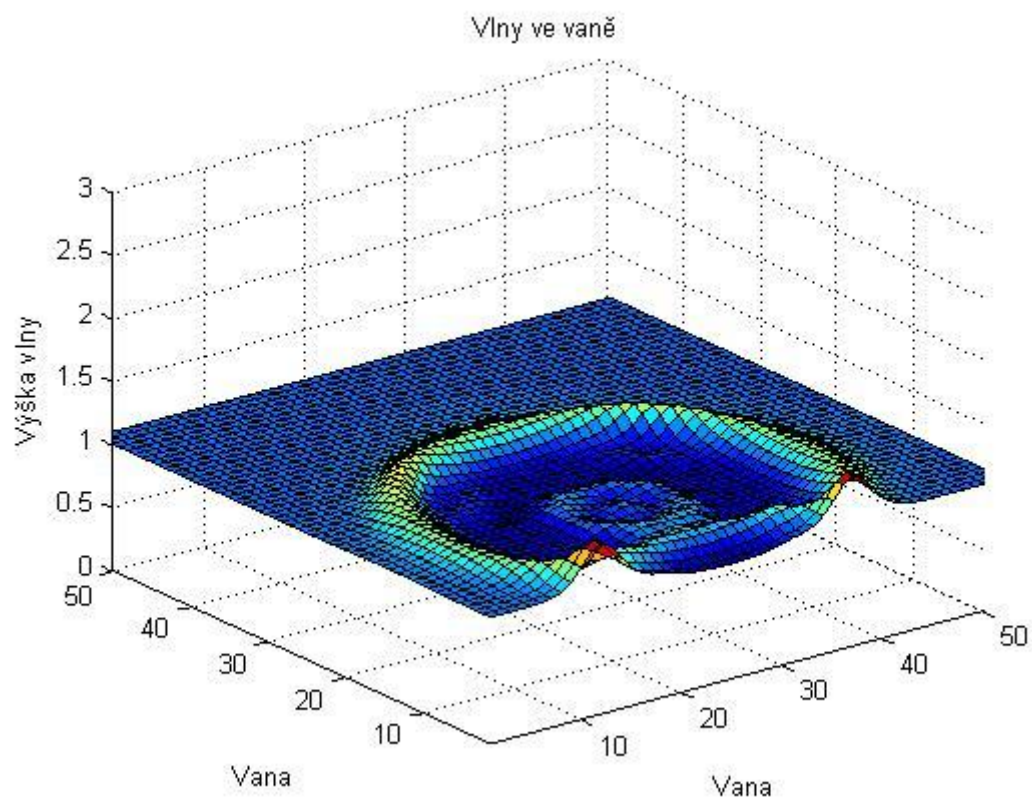
end



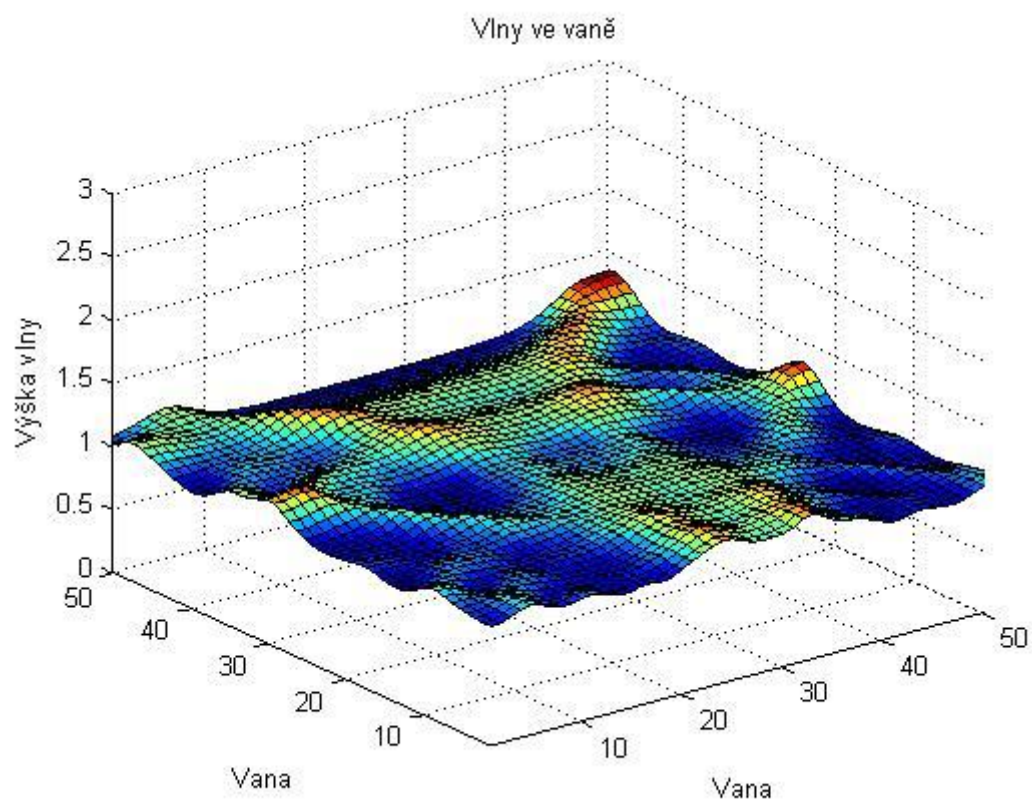
Obr. 1a: Simulovaná „kapka vody“ (počáteční vychýlení), která rozvlní hladinu.



Obr. 2a: Okamžik po dopadu kapky.



Obr. 3a: Vlny po nárazu na okraj vany.



Obr. 4a: Hladina po několika sekundách.

Cílem úlohy bylo simulovat vlnění hladiny ve čtvercové nádobě (vaně). Grafický výstup simulace se chová podle předpokladů a po zhroucení počáteční výchylky se začne šířit kruhová vlna, která se dále odráží od okrajů nádoby a ostatních vln. Jediný problém nastává při delším chodu programu, kdy nedochází k útlumu vlnění, ale naopak se vlnění zvětšuje. Zkoušel jsem různé velikosti sítě a integračního kroku. U sítě jsem zvolil takovou velikost, aby u grafického výstupu byla zachována jak dostatečná plynulost simulace, tak i dostatek detailů. U integračního kroku jsem zvolil kompromis mezi přesností a rychlostí.

Reference

[1] http://en.wikipedia.org/wiki/Shallow_water_equations

[2] http://en.wikipedia.org/wiki/Lax-Wendroff_method

2 Brownův pohyb

Tento jev je pojmenovaný po biologovi Robertu Brownovi. Brownův pohyb je náhodný pohyb částic umístěných v kapalině nebo plynu, který je způsoben srážkami částic s rychle se pohybujícími atomy kapaliny nebo plynu. Rychlost pohybu se zvyšuje s rostoucí teplotou a závisí na viskozitě látky a velikosti částic. Brownův pohyb má význam pro pochopení difúze látek v prostředí.

Náhodný charakter trajektorie částice znemožňuje určit měřením její časový průběh nebo používat pojem rychlost. Když bychom sledovaly pohyb částice zjišťováním okamžitých poloh, můžeme při dostatečném počtu měření získat střední hodnotu kvadrátu vzdálenosti, kterou částice urazí za daný časový interval. Pohyb sledujeme z bodu s polohovým vektorem $\mathbf{r} = 0$ do bodu s polohovým vektorem $\mathbf{r}(t)$ získáme tedy

$$\lambda(t) = \sqrt{\langle \mathbf{r}^2(t) \rangle}$$

Střední vzdálenost je přímo úměrná odmocnině z času t . Poté stačí pouze měřit průmět postupných poloh částice do jednoho směru, např. pro směr osy x měříme jen souřadnici $x(t)$ okamžité polohy a získáme tak

$$\lambda_x(t) = \sqrt{\langle x^2(t) \rangle} = \frac{\lambda(t)}{\sqrt{3}}$$

Rozdělovací funkci $f(x, t)$ použijeme pro sledování polohy částic v prostoru a má význam hustoty pravděpodobnosti polohy x částice v čase t .

Pro rozdělovací funkci musí platit

$$\int_{-\infty}^{+\infty} f(x, t) dx = 1$$

a jestliže je funkce známa tak nám umožní spočítat střední hodnoty všech veličin závislých na polohách částic. Potom tedy platí

$$\lambda_x(t) = \int_{-\infty}^{+\infty} x^2 f(x, t) dx$$

Zavedeme funkci $v(x, t) = Nf(x, t)$, která bude mít význam prostorové hustoty částic v časovém okamžiku t . Posouvání částic je výsledkem dynamické rovnováhy mezi působením osmotických sil a odporem prostředí. Pro odpor prostředí použijeme Stokesův vztah

$$F_{odp} = -6\pi\eta av$$

kde η je viskozita kapaliny, a je poloměr částice a v je rychlost částice. Zahrnutí důsledků obou vlivů vede na rovnici

$$\frac{df}{dt} = D \frac{d^2f}{dx^2}$$

kde D je koeficient difúze. Odvozený vztah použijeme k určení rozdělovací funkce $f(x, t)$. Je to rovnice difúze, z toho vyplývá, že rozptyl částic má difúzní charakter. Koeficient difúze lze vyjádřit následovně

$$D = \frac{k_B T}{6\pi\eta a}$$

kde $k_B = 1,381 \cdot 10^{-23} \text{ J} \cdot \text{K}^{-1}$ je Boltzmannova konstanta a T je absolutní teplota. Řešení difúzní rovnice má tvar

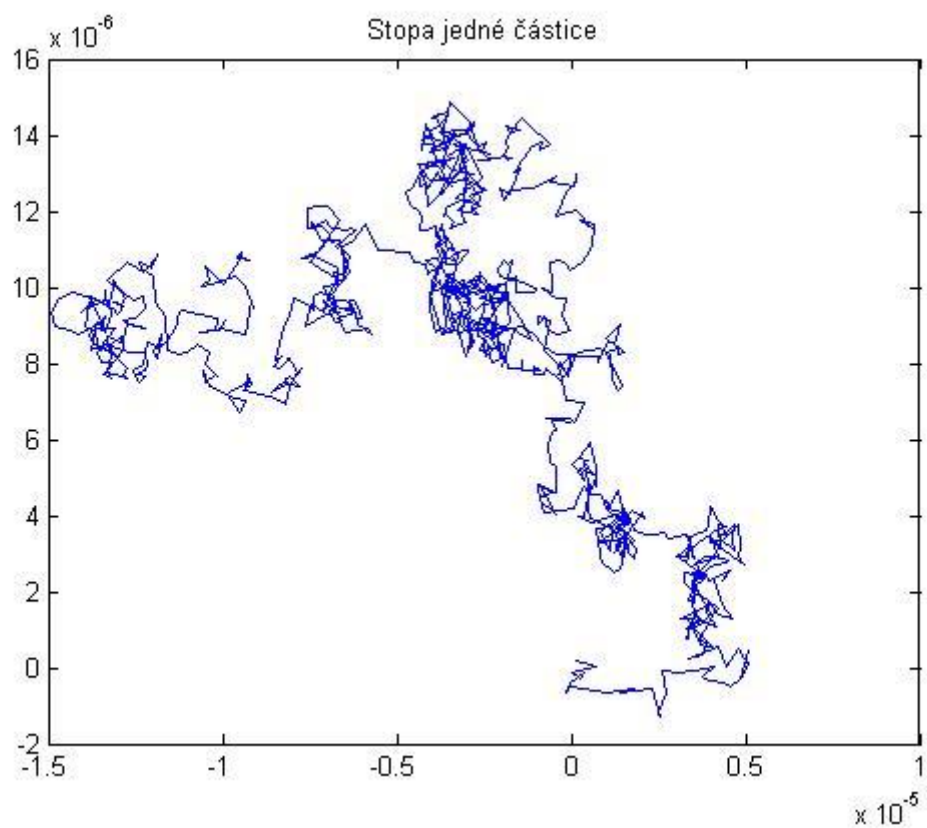
$$f(x, t) = \frac{1}{\sqrt{4\pi Dt}} e^{-\frac{x^2}{4Dt}}$$

Toto řešení odpovídá situaci, kdy všechny částice se začnou pohybovat z jednoho bodu ve stejný čas. Pro střední posuv brownovské částice dostaneme výsledek

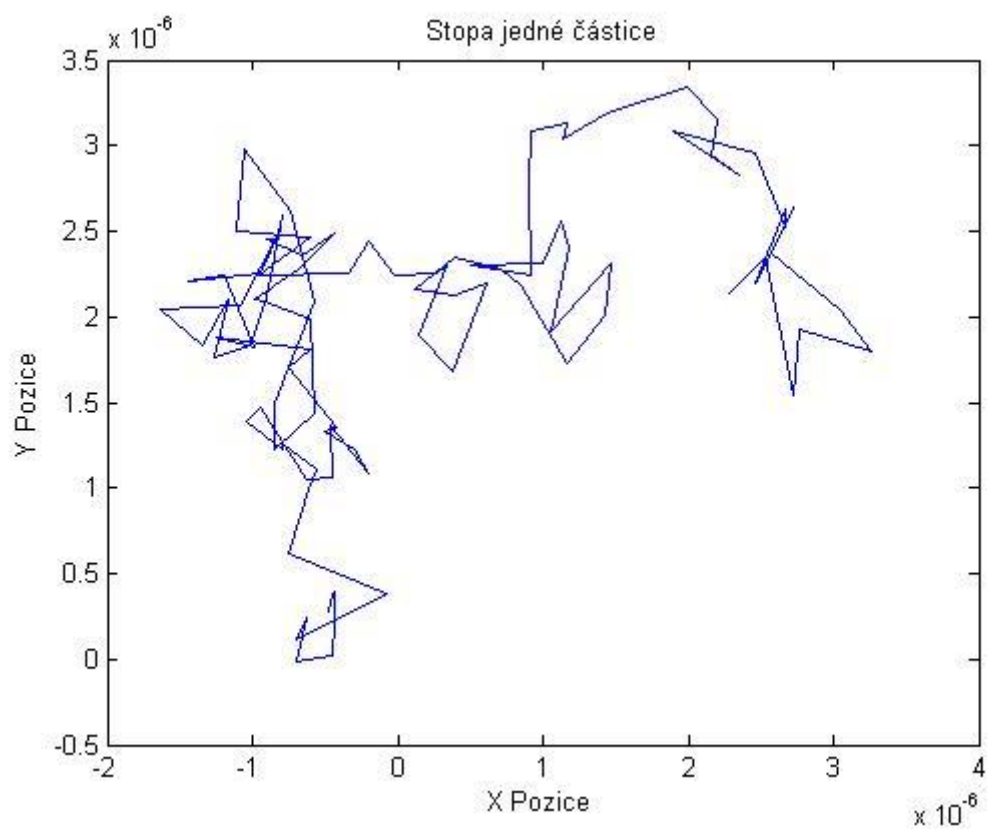
$$\lambda_x(t) = \sqrt{2Dt}$$

Zdrojový kód výpočtu v programu MATLAB

```
N = 1000;           % počet kroků
a = 0.5e-6;         % poloměr částice
eta = 1.0e-3;       % viskozita vody
kB = 1.38e-23;      % Boltzmannova konstanta
T = 293;            % teplota
D = kB * T / (6 * pi * eta * a)    % koeficient difúze
t = 0.1;
k = sqrt(2*D*t);
dx = k * randn(N,1);    dy = k * randn(N,1);
x = cumsum(dx);         y = cumsum(dy);
plot(x,y);
title('Stopa jedné částice');
ylabel('Y Pozice');
xlabel('X Pozice');
```



Obr. 1b: Simulace Brownova pohybu pro jednu částici pro 1000 kroků.



Obr. 2b: Simulace Brownova pohybu pro jednu částici pro 100 kroků.

Cílem úlohy bylo simulovat Brownův pohyb jedné částice ve vodě pomocí Einsteinova modelu. Grafický výstup programu je podle očekávání a vykresluje trajektorii částice ve 2D prostoru. Počet kroků byl zvolen tak aby byla dobře vidět náhodnost pohybu ale zároveň aby se stopa moc nepřekrývala.

Reference

[1] http://en.wikipedia.org/wiki/Brownian_motion

[2] http://dml.cz/bitstream/handle/10338.dmlcz/141268/PokrokyMFA_50-2005-3_1.pdf

3 Rovnice vedení tepla

Mějme dvourozměrnou rovnici vedení tepla s homogenními okrajovými podmínkami.

$$\frac{\partial u}{\partial t} = c^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$

Řešení rovnice s okrajovými podmínkami

$$u(0, y, t) = u(a, y, t) = 0, \quad 0 < y < b, \quad t > 0,$$

$$u(x, 0, t) = u(x, b, t) = 0, \quad 0 < x < a, \quad t > 0,$$

a počáteční podmínkou

$$u(x, y, 0) = f(x, y), \quad 0 < x < a, \quad 0 < y < b,$$

je

$$u(x, y, t) = \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} A_{mn} \sin \frac{m\pi}{a} x \sin \frac{n\pi}{b} y e^{-\lambda_{mn}^2 t}$$

kde

$$\lambda_{mn} = c\pi \sqrt{\frac{m^2}{a^2} + \frac{n^2}{b^2}}$$

a

$$A_{mn} = \frac{4}{ab} \int_0^b \int_0^a f(x, y) \sin \frac{m\pi}{a} x \sin \frac{n\pi}{b} y \, dx dy, \quad m, n = 1, 2, \dots$$

Řešme problém vedení tepla na čtvercové oblasti s $a = b = 1$, a $c = 1/\pi$.

Předpokládejme, že okraje jsou udržovány na nulové teplotě a počáteční rozdělení teploty je $u(x, y, 0) = 100^\circ$.

Řešení: Máme

$$u(x, y, t) = \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} A_{mn} \sin m\pi x \sin n\pi y e^{-\lambda_{mn}^2 t}$$

kde

$$\lambda_{mn} = \sqrt{m^2 + n^2}$$

a

$$A_{mn} = 400 \int_0^1 \int_0^1 \sin m\pi x \sin n\pi y \, dx dy = \frac{400}{\pi^2} \frac{[1 - (-1)^m][1 - (-1)^n]}{mn}$$

Protože, A_{mn} je nula jestliže buď m nebo n je sudé, my dostaneme

$$u(x, y, t) = \frac{1600}{\pi^2} \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \frac{\sin(2l+1)\pi x \sin(2k+1)\pi y}{(2l+1)(2k+1)} e^{-\lambda_{(2l+1)(2k+1)}^2 t}$$

Zdrojový kód výpočtu v programu MATLAB

```
clear all;
```

```
close all;
```

```
x = (0:0.02:1);
```

```
y = (0:0.02:1);
```

```
[X,Y] = meshgrid(x,y);
```

```
nt = 50;           % počet časových kroků
```

```
dt = 0.05;        % integrační krok
```

```
N = 25;
```

```
c = colormap('jet');
```

```
s = surf('cDataMapping','direct','cData',c);
```

```
axis([0 1 0 1 0 100]);
```

```
title('Chladnutí desky');
```

```
xlabel('Deska');
```

```
ylabel('Deska');
```

```
zlabel('Teplota [°]');
```

```
% Časová smyčka
```

```
for j = 1:nt
```

```
    u = zeros(length(x),length(y));
```

```
    % Dvojitá suma
```

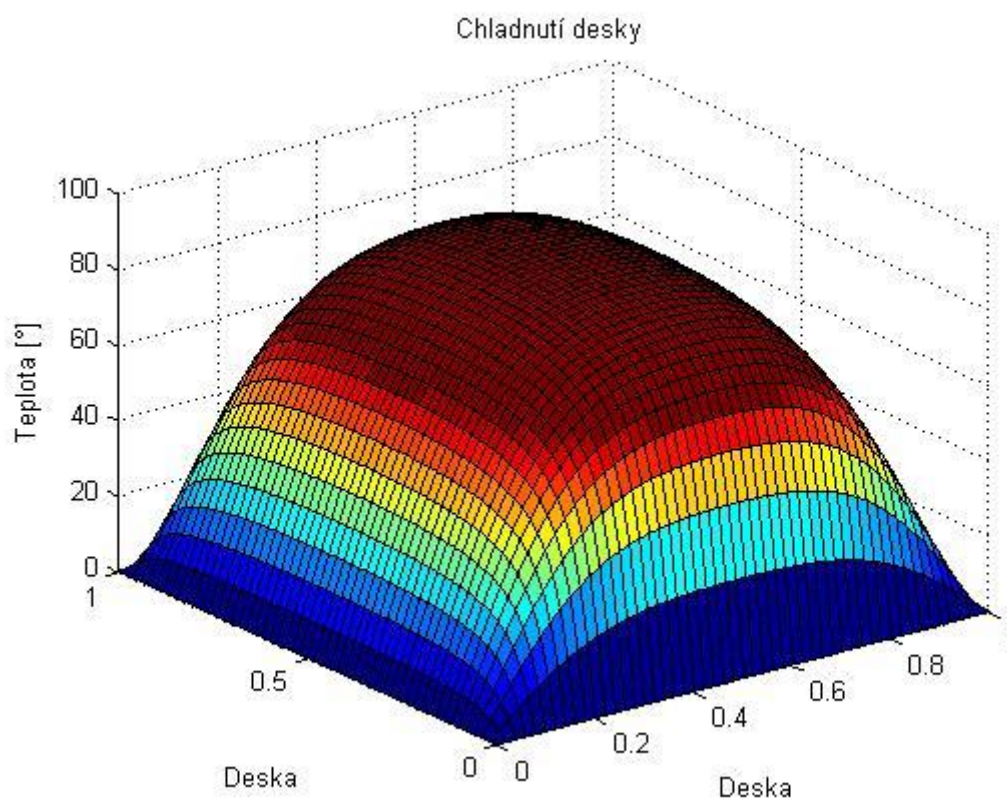
```
    for m = 1:N      % index on x
```

```
        for n = 1:N  % index on y
```

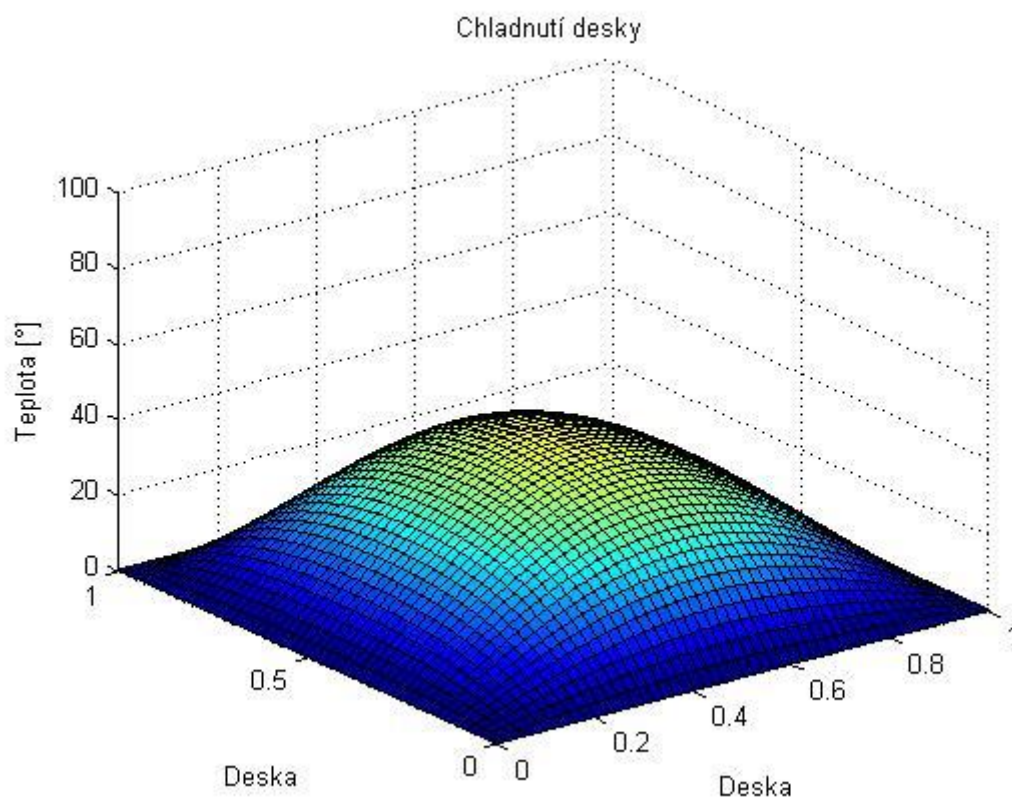
```

Bmn = 1600 / (pi^2*(2*m-1)*(2*n-1));
u = u + Bmn * sin((2*m-1)*pi*X) .* sin((2*n-1)*pi*Y) *...
    exp(-(2*m-1)^2 + (2*n-1) ) * (j*dt));
end
end
set(s,'xdata',x,'ydata',y,'zdata',u)
drawnow
end

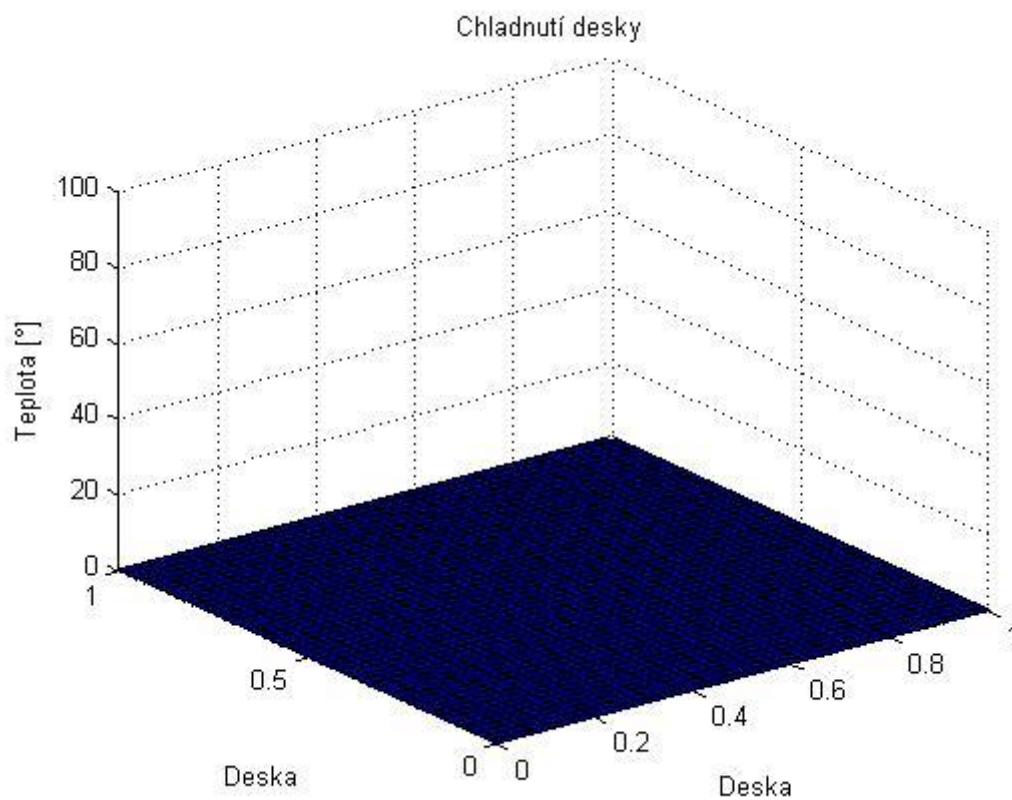
```



Obr. 1c: Zahřátá deska na počátku.



Obr. 2c: Deska po chvíli chladnutí.



Obr. 3c: Vychladnutá deska.

Cílem úlohy bylo simulovat chladnutí homogenní desky, kdy okraje měli konstantní teplotu (0°C) a zbytek desky byl zahřátý na jinou teplotu (100°C). Z grafického výstupu programu je vidět postupné chladnutí desky až do okamžiku úplného vychladnutí, ve kterém má celá deska stejnou teplotu.

Reference

[1] Nakhle H. Asmar: *Partial Differential Equations with Fourier Series and Boundary Value Problems (2nd Edition)*, Pearson, (2004), 816 s.