DOKTORANDSKÉ DNY 2009

sborník workshopu doktorandů FJFI
oboru Matematické inženýrství

13. a 20. listopadu 2009

P. Ambrož, Z. Masáková (editoři)

Doktorandské dny 2009
sborník workshopu doktorandů FJFI oboru Matematické inženýrství

P. Ambrož, Z. Masáková (editoři)
Kontakt petr.ambroz@fjfi.cvut.cz / 224 358 569

Vydalo České vysoké učení technické v Praze
Zpracovala Fakulta jaderná a fyzikálně inženýrská
Vytisklo Nakladatelství ČVUT-výroba, Zikova 4, Praha 6
Počet stran 176, Vydání 1.

ISBN 978-80-01-04436-0

Seznam příspěvků

Předmluva

Workshop Doktorandské dny 2009 je již čtvrtým v řadě setkání doktorandů oboru Matematické inženýrství doktorského studijního programu Aplikace přírodních věd, který je akreditovaný na katedrách matematiky a fyziky FJFI. Pro mnohé z doktorandů je tento workshop první příležitostí referovat o své vědecké práci před odborným fórem. Nácvik prezentačních dovedností je pro postgraduální studenty nezbytný, stejně jako zkušenost s psaním odborných textů. I z tohoto důvodu jsou příspěvky doktorandů shromážděny v psané formě v tomto sborníku. Sborníky Doktorandských dnů pak slouží i ke sledování postupu práce jednotlivých doktorandů. Otištěné práce mají obvykle vysokou úroveň a nezřídka bývají výsledky našich doktorandů později publikovány v recenzovaných odborných časopisech..

Za přispění ke zdárnému konání workshopu děkujeme katedře matematiky FJFI, kde se setkání koná, i Dopplerovu ústavu pro matematickou fyziku a aplikovanou matematiku při FJFI, který konání finančně podpořil.

Editoři

Paralelní algoritmy pro numerické řešení hydrodynamiky laserového plazmatu

Ľuboš Bednárik

2. ročník PGS, email: Lbs@centrum.sk
Katedra matematiky, Fakulta jadrová a fyzikálne inžinierska, ČVUT v Praze
školiteľ: Richard Liska*, Katedra fyzikální elektroniky, Fakulta jaderná
a fyzikálně inženýrská, ČVUT

Abstract. For solution of laser plasma hydrodynamic we introduce model of Lagrangian equations, which includes heat conductivity and laser absorption. We show us the discretization of hydrodynamical equations and describe one step of the difference schema. With achieved results we check the correctness of our solution.

Abstrakt. Pre riešenie hydrodynamiky laserovej plazmy sa v úvode zoznámime s modelom Lagrangeovských rovníc, ktorý v sebe zahŕňa aj tepelnú vodivosť a laserovú absorpciu. Ukážeme si diskretizáciu hydrodynamických rovníc a popíšeme jeden cyklus diferenčnej schémy. Získanými výsledkami overíme korektnosť nášho riešenia.

1 Formulácia úlohy

Laserová plazma, ktorá vzniká pri interakcii laserového žiarenia s hmotou, je typicky modelovaná ako stlačiteľná kvapalina prostredníctvom Eulerových rovníc s tepelnou vodivosťou a laserovou absorpciou. Simuláciou vznikajú oblasti, ktoré sa vyznačujú vysokou expanziou resp. kompresiou. Popis systému v Lagrangeovských súradniciach je preto vhodnejší než klasický Eulerovský popis, ktorý nie je vhodný pre problémy, kde nastávajú veľké zmeny vo výpočtovej doméne (podrobný popis transformácie môžeme nájsť v [6, 7]). Budeme sa teda venovať problému, ktorý v Lagrangeovských súradniciach (S, t) má tvar

$$\frac{d\eta}{dt} = v_S \tag{1}$$

$$\frac{dv}{dt} = -p_S \tag{2}$$

$$\frac{d\varepsilon}{dt} = -pv_S - W_S - L_S \tag{3}$$

kde $\eta = 1/\rho$, ρ je hustota, v rýchlosť, p tlak, ε vnútorná energia, W je tepelný tok a L je hustota toku energie (intenzita) laserového žiarenia. Jednotlivé rovnice vyjadrujú postupne zákon zachovania hmotnosti (1), zákon zachovania hybnosti (2) a zákon zachovania energie (3). Systém doplňujeme ďalej ešte o stavové rovnice $p = p(\varepsilon, \rho)$, $T = T(\varepsilon, \rho)$,

[*] liska@siduri.fjfi.cvut.cz

ktoré pre ideálny plyn uvažujeme v tvare:

$$p = \varepsilon\rho(\gamma - 1) \quad (4)$$

$$T = \frac{A}{Z+1}\frac{p}{c_p\rho},\ c_p = \frac{k_B}{m_u} \quad (5)$$

kde $\gamma = 5/3$ je plynová konštanta, Z stupeň ionizácie, A atómové číslo, k_B Boltzmanova konštanta a $m_u = 1,6605.10^{-24}g$ atómová hmotnostná jednotka.

Systém rovníc (1), (2), (3) riešime v dvoch krokoch. V prvom kroku riešime samostatne systém hydrodynamických rovníc

$$\frac{d\eta}{dt} = v_S \quad (6)$$

$$\frac{dv}{dt} = -p_S \quad (7)$$

$$\frac{d\varepsilon}{dt} = -pv_S \quad (8)$$

V druhom kroku riešime samostatne rovnicu vedenia tepla so zahrnutým členom pre laserove žiarenie

$$\frac{d\varepsilon}{dt} = -W_S - L_S \quad (9)$$

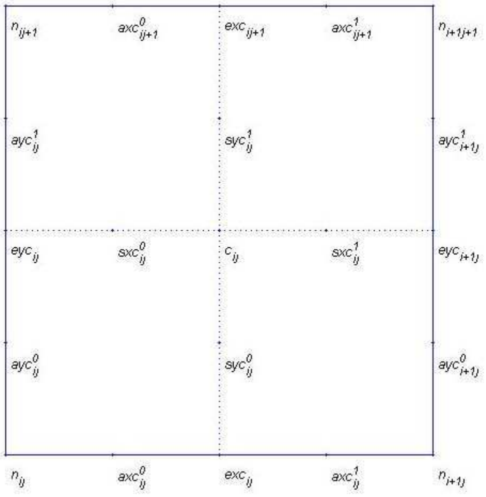
2 Diskretizácia

Systém riešime numericky diskretizáciou v čase aj v priestore, pričom uvažujeme obdĺžnikovú doménu $\langle A^x, B^x\rangle \times \langle A^y, B^y\rangle$. Táto oblasť je ľubovoľne rozdelená bodmi n_{11} až $n_{m^x+1m^y+1}$ na $m^x m^y$ buniek, kde $n_{11} = (A^x, A^y)$ a $n_{m^x+1m^y+1} = (B^x, B^y)$. Tieto bunky tvoria takzvanú *primárnu sieťku. Primárne body* definujeme ako stredy týchto buniek a značíme postupne $n_{1/2,1/2}$ až $n_{m^x+1/2,m^y+1/2}$ resp. c_{11} až $c_{m^x m^y}$, kde $c_{ij} = n_{i+1/2,j+1/2}$ je stred bunky definovanej vrcholmi n_{ij},n_{i+1j},n_{i+1j+1},n_{ij+1}. Vrcholy primárnej sieťky tvoria tzv. *duálne body* a *duálna sieťka* bude obsahovať duálne body vnútri svojich buniek, a teda jej vrcholmi sú primárne body.

Bunky a uzly na okraji domény nazývame *okrajové bunky* a *okrajové uzly.* Po okraji celej domény navyše pridávame ešte jednu vrstvu uzlov, tzv. *ghost uzly*, ktoré spolu s okrajovými uzlami vytvárajú *ghost bunky.* Polohy ghost uzlov sú rovnaké ako polohy okrajových uzlov, z čoho vyplýva, že ghost bunky majú nulový objem. Pri zmene polohy okrajových uzlov sa analogicky zmení poloha ghost uzlov tak, aby sa udržala nulovosť objemov ghost buniek. Význam ghost buniek a ghost uzlov sa uplatňuje pri definícii a aplikácii okrajovej podmienky.

Na obrázku (1) môžme vidieť zobrazenú ij-tu bunku tvorenú 4 vrcholmi primárnej sieťky. V jej strede sa nachádza 1 bod duálnej sieťky. Stredy hrán sú označené písmenamy *exc* a *eyc* podľa toho, či ide o horizontálnu hranu orientovanú v smere osi x alebo vertikálnu hranu orientovanú v smere osi y. Spojnicu stredu hrany a stredu bunky nazývame *separátor.* Každá bunka má teda 4 separátory, dva horizontálne sxc^0_{ij},sxc^1_{ij} a dva vertikálne syc^0_{ij}, syc^1_{ij}.

Každá hrana je navyše svojím stredom rozdelená na dve subhrany, ktoré značíme axc^0_{ij},axc^1_{ij} a ayc^0_{ij},ayc^1_{ij}. Tým je celá bunka rozdelená na štvoricu subzón, kde každa



Obrázok 1: Štruktúra ij-tej bunky.

je tvorená dvojícou príslušných subhrán a dvojícou príslušných separátorov. Indexácia subzón je v poradí zľava zdola 00, 10, 11, 01 proti smeru hodinových ručičiek.

Ďalej potrebujeme určiť objem bunky, prípadne jednotlivých subzón, ktorý závisí na použitom type geometrii. Budeme uvažovať karteziánsku a cylindrickú geometriu, ktoré si ďalej podrobnejšie rozoberieme. Budeme pritom potrebovať znalosť greenovej vety

$$\int\limits_V \frac{\partial A}{\partial x} dxdy - \frac{\partial B}{\partial y} dxdy = \int\limits_{\partial V} Adx + Qdy \tag{10}$$

2.1 Tlakové sily v karteziánskej geometria

V prípade karteziánskej geometrie uvažujeme nasledujúci vzťah pre vypočet objemu bunky

$$V_c = \int\limits_c 1dxdy. \tag{11}$$

S použitím Greenovej vety objemový integrál sa zmení na krivkový integrál po hranici

$$V_c = \int\limits_{\partial V_c} xdy = \int\limits_{n_{ij}}^{n_{i+1j}} xdy + \int\limits_{n_{i+1j}}^{n_{i+1j+1}} xdy + \int\limits_{n_{i+1j+1}}^{n_{ij+1}} xdy + \int\limits_{n_{ij+1}}^{n_{ij}} xdy,$$

a v prípade, že označíme vrcholy bunky postupne číslami od 1 do 4 dostávame

$$V_c = \sum_{l=1}^{4} \left(y_{l+1} - y_l\right)\left(x_l + x_{l+1}\right).$$

Analogicky dokážeme spočítať objem subzóny a následne spolu s hustotou subzóny získavame hmotnosť subzóny

$$m_c^l = \rho_c^l V_c^l.$$

Pre celkovú hmotnosť a celkový objem bunky pritom prirodzene platí

$$m_c = \sum_{l=1}^{4} m_c^l, \quad V_c = \sum_{l=1}^{4} V_c^l.$$

Hmotnosť uzlu ako aj objem uzlu definujeme opäť prirodzeným spôsobom ako súčet hmotností resp. objemov subzón okolo daného uzlu.

Integráciou rovnice pre zákon zachovania hybnosti dostávame

$$\rho\frac{dw^x}{dt} = -\frac{\partial p}{\partial x}\Big/ . \int\limits_{V_n} dxdy \quad \Rightarrow \quad \int\limits_{V_n} \rho\frac{dw^x}{dt}dxdy = m_n\left(\frac{dw^x}{dt}\right)_n = -\int\limits_{V_n}\frac{\partial p}{\partial x}dxdy = F_{pn}^x \tag{12}$$

$$\rho\frac{dw^y}{dt} = -\frac{\partial p}{\partial y}\Big/ . \int\limits_{V_n} dxdy \quad \Rightarrow \quad \int\limits_{V_n} \rho\frac{dw^y}{dt}dxdy = m_n\left(\frac{dw^y}{dt}\right)_n = -\int\limits_{V_n}\frac{\partial p}{\partial y}dxdy = F_{pn}^y \tag{13}$$

kde horným indexom x resp. y máme na mysli príslušnu zložku danej veličiny a symbolom F_{pn} označujeme tlakovú silu pôsobiacu na uzol n. Následne s použitím Greenovej vety dostaneme

$$F_{pn}^x \overset{(10)}{=} -\int\limits_{\partial V_n} pdy \tag{14}$$

$$F_{pn}^y \overset{(10)}{=} +\int\limits_{\partial V_n} pdx \tag{15}$$

Hranica uzlového objemu je tvorená ôsmimi separatormi v okolí daného uzlu n, ktoré označíme cyklicky číslami od 0 do 7 zľava zdola spodným separátorom počínajúc proti smeru hodinových ručičiek. Separátor číslo sedem je zároveň mínus prvým separátorom. Môžme tak napísať vzťah

$$\partial V_n = \sum_{l=0}^{7} s(n,l) = \sum_{l=-1}^{6} s(n,l) = \sum_{l=0}^{3}\sum_{k=0}^{1} s(n, 2l+k-1) \tag{16}$$

Naše separátory sú vždy orientované v smere doprava a nahor, preto zavádzame tzv. znamienkový integrál po separátore

$$\widetilde{\int\limits_{s(n,l)}} dy = Sgn\left(s(n,l)\right) \int\limits_{s(n,l)} dy \tag{17}$$

$$\widetilde{\int\limits_{s(n,l)}} dx = Sgn\left(s(n,l)\right) \int\limits_{s(n,l)} dx \tag{18}$$

ktorý v prípade, že budeme integrovať po danom separátore v smere proti jeho orientácii má opačné znamienko ako integrál po tomto separátore. V opačnom prípade má znamienko rovnaké. Tým docielime istú kompaktnosť finálnych vzťahv pre výpočet tlakovej sily pôsobiacej na daný uzol.

Podľa zadefinovaného značenia teda platí

$$F_{pn}^{x} \overset{(16)+(17)}{=} -\sum_{l=-1}^{6} p_{s(n,l)} \widetilde{\int\limits_{s(n,l)}} dy = -\sum_{l=0}^{3}\sum_{k=0}^{1} p_{s(n,2l+k-1)} \widetilde{\int\limits_{s(n,2l+k-1)}} dy \tag{19}$$

$$F_{pn}^{y} \overset{(16)+(18)}{=} +\sum_{l=-1}^{6} p_{s(n,l)} \widetilde{\int\limits_{s(n,l)}} dx = +\sum_{l=0}^{3}\sum_{k=0}^{1} p_{s(n,2l+k-1)} \widetilde{\int\limits_{s(n,2l+k-1)}} dx \tag{20}$$

Tlakom na separátore rozumieme priemer tlakov subzón susediacich s týmto separátorom. Ak uvažujeme, že subzóny v bunke máme očíslované zľava zdola číslami 0 až 3 a podobne aj bunky v okolí uzlu n máme očíslované zľava zdola číslami od 0 do 3, potom označením $p^{m}_{c(n,l)}$ rozumieme tlak v m-tej subzóne l-tej bunky pri uzle n. Následne teda tlak na $2l+k-1$-tom separátore (číslovanie separátorov zo vzťahu 16) sa dá vyjadriť ako

$$p_{s(n,2l+k-1)} = \frac{1}{2}\left(p^{l+2-k}_{c(n,l)} + p^{l+2-k+1}_{c(n,l)}\right) = \frac{1}{2}\left(p^{l-k+2}_{c(n,l)} + p^{l-k+3}_{c(n,l)}\right) \tag{21}$$

Pre zložky tlakových síl potom platí

$$F_{pn}^{x} \overset{(21)}{=} -\sum_{l,k=0}^{3,1} \frac{p^{l-k+2}_{c(n,l)} + p^{l-k+3}_{c(n,l)}}{2} \widetilde{\int\limits_{s(n,2l+k-1)}} dy \tag{22}$$

$$F_{pn}^{y} \overset{(21)}{=} +\sum_{l,k=0}^{3,1} \frac{p^{l-k+2}_{c(n,l)} + p^{l-k+3}_{c(n,l)}}{2} \widetilde{\int\limits_{s(n,2l+k-1)}} dx \tag{23}$$

S týmito vzťahmi by sme v podstate mohli aj skončiť, avšak aby sme dosiahli podobnosť so vzťahmi pre cylindricku geometriu, vykonáme ešte niekoľko formálnych úprav. K tomu si potrebujeme vyjadriť orientovanú hranicu subzonálneho objemu

$$\partial V^{l+2}_{c(n,l)} = s(n,2l-1)+s(n,2l)+a(n,l)-a(n,l-1) = \sum_{k=0}^{1}\big(s(n,2l+k-1)+(-1)^{k}a(n,l-k)\big) \tag{24}$$

kde písmenom a označujeme príslušné subhrany. Inými slovami orientovaná hranica $l+2$-tej subzóny l-tej bunky pri uzle n je tvorená $2l-1$-tým a $2l$-tým separátorom pri uzle n a l-tou a $l-1$-tou subhranou pri uzle n. Mínus pred $l-1$-tou hranou nám hovorí, že táto hrana je opačne orientovaná než je orientácia hranice na tomto úseku. Číslovanie subhrán v okolí uzlu n začína dolnou subhranou od 0 do 3. Podobne ako sme zaviedli znamienkový integrál po separátore, zavádzame aj znamienkový integrál po subhrane

$$\widetilde{\int\limits_{a(n,l)}} dy = Sgn\,(a(n,l)) \int\limits_{a(n,l)} dy \tag{25}$$

$$\widetilde{\int\limits_{a(n,l)}} dx = Sgn\,(a(n,l)) \int\limits_{a(n,l)} dx \tag{26}$$

Keďže hranice subzón sú uzavreté krivky, platí:

$$0 = \int\limits_{\partial V^{l+2}_{c(n,l)}} dy = p^{l+2}_{c(n,l)} \int\limits_{\partial V^{l+2}_{c(n,l)}} dy \overset{(24)+(25)}{=} p^{l+2}_{c(n,l)} \sum_{k=0}^{1} \widetilde{\int\limits_{s(n,2l+k-1)}} dy + p^{l+2}_{c(n,l)} \sum_{k=0}^{1}(-1)^{k} \widetilde{\int\limits_{a(n,l-k)}} dy \tag{27}$$

$$0 = \int\limits_{\partial V^{l+2}_{c(n,l)}} dx = p^{l+2}_{c(n,l)} \int\limits_{\partial V^{l+2}_{c(n,l)}} dx \overset{(24)+(26)}{=} p^{l+2}_{c(n,l)} \sum_{k=0}^{1} \widetilde{\int}\limits_{s(n,2l+k-1)} dx + p^{l+2}_{c(n,l)} \sum_{k=0}^{1} (-1)^k \widetilde{\int}\limits_{a(n,l-k)} dx \tag{28}$$

a po pridaní do vzťahov pre tlakove sily získame:

$$F^x_{pn} \overset{(27)}{=} -\sum_{l,k=0}^{3,1} \frac{p^{l-k+2}_{c(n,l)} + p^{l-k+3}_{c(n,l)}}{2} \widetilde{\int}\limits_{s(n,2l+k-1)} dy + \sum_{l,k=0}^{3,1} p^{l+2}_{c(n,l)} \widetilde{\int}\limits_{s(n,2l+k-1)} dy + \sum_{l,k=0}^{3,1} p^{l+2}_{c(n,l)} (-1)^k \widetilde{\int}\limits_{a(n,l-k)} dy \tag{29}$$

$$F^y_{pn} \overset{(28)}{=} +\sum_{l,k=0}^{3,1} \frac{p^{l-k+2}_{c(n,l)} + p^{l-k+3}_{c(n,l)}}{2} \widetilde{\int}\limits_{s(n,2l+k-1)} dx - \sum_{l,k=0}^{3,1} p^{l+2}_{c(n,l)} \widetilde{\int}\limits_{s(n,2l+k-1)} dx - \sum_{l,k=0}^{3,1} p^{l+2}_{c(n,l)} (-1)^k \widetilde{\int}\limits_{a(n,l-k)} dx \tag{30}$$

2.2 Tlakové sily v cylindrickej geometrii

V prípade cylindrickej geometrie vzťah pre vypočet objemu bunky vyzerá nasledujúco:

$$V_c = \int\limits_c 1rdrdz. \tag{31}$$

S použitím Greenovej vety sa nám objemový integrál opäť zredukuje na krivkový integrál

$$V_c = \sum_{e \in \partial c} \int\limits_e \frac{r^2}{2} dz = \frac{1}{6} \sum_{l=1}^{4} \left(z_{l+1} - z\right) \left(r_l^2 + r_{l+1}^2 + r_l r_{l+1}\right).$$

Integráciou rovnice pre zákon zachovania hybnosti dostaneme v cylindrickej geometrii vzťahy pre zložky tlakových síl

$$\rho \frac{dw^r}{dt} = -\frac{\partial p}{\partial r} \Big/ . \int\limits_{V_n} rdrdz \quad \Rightarrow \quad \int\limits_{V_n} \rho \frac{dw^r}{dt} rdrdz = m_n \left(\frac{dw^r}{dt}\right)_n = -\int\limits_{V_n} \frac{\partial p}{\partial r} rdrdz = F^r_{pn} \tag{32}$$

$$\rho \frac{dw^z}{dt} = -\frac{\partial p}{\partial z} \Big/ . \int\limits_{V_n} rdrdz \quad \Rightarrow \quad \int\limits_{V_n} \rho \frac{dw^z}{dt} rdrdz = m_n \left(\frac{dw^z}{dt}\right)_n = -\int\limits_{V_n} \frac{\partial p}{\partial z} rdrdz = F^z_{pn} \tag{33}$$

Pre derivácie tlaku odvodíme

$$\frac{\partial (pr)}{\partial r} = \frac{\partial p}{\partial r} r + p \frac{\partial r}{\partial r} = \frac{\partial p}{\partial r} r + p \quad \Rightarrow \quad \frac{\partial p}{\partial r} r = \frac{\partial (pr)}{\partial r} - p \frac{\partial r}{\partial r} = \frac{\partial (pr)}{\partial r} - p \tag{34}$$

$$\frac{\partial (pr)}{\partial z} = \frac{\partial p}{\partial z} r + p \frac{\partial r}{\partial z} = \frac{\partial p}{\partial z} r + 0 \quad \Rightarrow \quad \frac{\partial p}{\partial z} r = \frac{\partial (pr)}{\partial z} - p \frac{\partial r}{\partial z} = \frac{\partial (pr)}{\partial z} - 0 \tag{35}$$

Po dosadení do vzťahov pre r-ovú a z-ovú zložku tlakovej sily dostávame tieto v podobe

$$F^r_{pn} \overset{(34)}{=} -\int\limits_{V_n} \frac{\partial (pr)}{\partial r} drdz + \int\limits_{V_n} p \frac{\partial r}{\partial r} drdz = -\int\limits_{V_n} \frac{\partial (pr)}{\partial r} drdz + \sum_{l=0}^{3} p^{l+2}_{c(n,l)} \int\limits_{V^{l+2}_{c(n,l)}} \frac{\partial r}{\partial r} drdz \tag{36}$$

$$F^z_{pn} \overset{(35)}{=} -\int\limits_{V_n} \frac{\partial\,(pr)}{\partial z} drdz + \int\limits_{V_n} p\frac{\partial r}{\partial z} drdz = -\int\limits_{V_n} \frac{\partial\,(pr)}{\partial z} drdz + \sum_{l=0}^{3} p^{l+2}_{c(n,l)} \int\limits_{V^{l+2}_{c(n,l)}} \frac{\partial r}{\partial z} drdz \tag{37}$$

čo spolu s Greenovou vetou dáva

$$F^r_{pn} \overset{(10)}{=} -\int\limits_{\partial V_n} prdz + \sum_{l=0}^{3} p^{l+2}_{c(n,l)} \int\limits_{\partial V^{l+2}_{c(n,l)}} rdz \tag{38}$$

$$F^z_{pn} \overset{(10)}{=} +\int\limits_{\partial V_n} prdr - \sum_{l=0}^{3} p^{l+2}_{c(n,l)} \int\limits_{\partial V^{l+2}_{c(n,l)}} rdr \tag{39}$$

Tu je vhodné poznamenať, že zatiaľ čo druhý člen v (38) je nulový, v (39) može byť druhý člen nenulový. V tom sa rozchádza analógia s karteziánskou geometriou, a práve možná nenulovosť druhého členu v (39) bola dôvodom dodatočných úprav v karteziánskom prípade, aby sme zachovali kompaktnosť konečných vzťahov pre tlakové sily.

Ďalej teda po zavedení znamienkových integrálov po separatore a subhrane v cylindrickom prípade

$$\widetilde{\int\limits_{s(n,l)}} rdz = Sgn\,(s(n,l)) \int\limits_{s(n,l)} rdz \tag{40}$$

$$\widetilde{\int\limits_{s(n,l)}} rdr = Sgn\,(s(n,l)) \int\limits_{s(n,l)} rdr \tag{41}$$

$$\widetilde{\int\limits_{a(n,l)}} rdz = Sgn\,(a(n,l)) \int\limits_{a(n,l)} rdz \tag{42}$$

$$\widetilde{\int\limits_{a(n,l)}} rdr = Sgn\,(a(n,l)) \int\limits_{a(n,l)} rdr \tag{43}$$

a po následných úpravách získavame

$$F^r_{pn} \overset{(21)}{=} -\sum_{l,k=0}^{3,1} \frac{p^{l-k+2}_{c(n,l)} + p^{l-k+3}_{c(n,l)}}{2} \widetilde{\int\limits_{s(n,2l+k-1)}} rdz + \sum_{l,k=0}^{3,1} p^{l+2}_{c(n,l)} \widetilde{\int\limits_{s(n,2l+k-1)}} rdz + \sum_{l,k=0}^{3,1} p^{l+2}_{c(n,l)} (-1)^k \widetilde{\int\limits_{a(n,l-k)}} rdz \tag{44}$$

$$F^z_{pn} \overset{(21)}{=} +\sum_{l,k=0}^{3,1} \frac{p^{l-k+2}_{c(n,l)} + p^{l-k+3}_{c(n,l)}}{2} \widetilde{\int\limits_{s(n,2l+k-1)}} rdr - \sum_{l,k=0}^{3,1} p^{l+2}_{c(n,l)} \widetilde{\int\limits_{s(n,2l+k-1)}} rdr - \sum_{l,k=0}^{3,1} p^{l+2}_{c(n,l)} (-1)^k \widetilde{\int\limits_{a(n,l-k)}} rdr \tag{45}$$

Zavedením zovšeobecneného značenia súradníc

$$Q = (q_0, q_1, q_2, q_3) = (x, y, r, z) \tag{46}$$

a perznačením všetkých integrálov nasledujúcim spôsobom

$$I^{q_0}_{s(n,l)} = I^{x}_{s(n,l)} = \underset{s(n,l)}{\widetilde{\int}} dy \qquad I^{q_0}_{a(n,l)} = I^{x}_{a(n,l)} = \underset{a(n,l)}{\widetilde{\int}} dy \tag{47}$$

$$I^{q_1}_{s(n,l)} = I^{y}_{s(n,l)} = \underset{s(n,l)}{\widetilde{\int}} dx \qquad I^{q_1}_{a(n,l)} = I^{y}_{a(n,l)} = \underset{a(n,l)}{\widetilde{\int}} dx \tag{48}$$

$$I^{q_2}_{s(n,l)} = I^{r}_{s(n,l)} = \underset{s(n,l)}{\widetilde{\int}} rdz \qquad I^{q_2}_{a(n,l)} = I^{r}_{a(n,l)} = \underset{a(n,l)}{\widetilde{\int}} rdz \tag{49}$$

$$I^{q_3}_{s(n,l)} = I^{z}_{s(n,l)} = \underset{s(n,l)}{\widetilde{\int}} rdr \qquad I^{q_3}_{a(n,l)} = I^{z}_{a(n,l)} = \underset{a(n,l)}{\widetilde{\int}} rdr \tag{50}$$

získavame zovšeobecnenú zložku tlakovej sily v tvare

$$\begin{aligned} F^{q_i}_{pn} = &-(-1)^i \sum_{l,k=0}^{3,1} \frac{1}{2} \left(p^{l-k+2}_{c(n,l)} + p^{l-k+3}_{c(n,l)} \right) I^{q_i}_{s(n,2l+k-1)} + \\ &+ (-1)^i \sum_{l,k=0}^{3,1} p^{l+2}_{c(n,l)} I^{q_i}_{s(n,2l+k-1)} + (-1)^i \sum_{l,k=0}^{3,1} p^{l+2}_{c(n,l)} (-1)^k I^{q_i}_{a(n,l-k)} \end{aligned} \tag{51}$$

kde $i = \{0, 1, 2, 3\}$.

2.3 Viskózna sila

Viskózna sila je veľmi dôležitou časťou celkovej sily pôsobiacej na uzol. Bez nej nie je Lagrangeovský riešič schopný simulovať problémy, v ktorých dochádza k rázovým vlnám a kontakným diskontinuitám. Existuje mnoho spôsobov ako zahrnúť viskoznu silu do riešenia. My využívame jeden z jednoduchších spôsobov, kde na výpočet viskóznej sily použijeme vzťahy získané pre tlakové sily, pričom miesto tlaku dosadíme do týchto vzťahov umelú viskozitu. Tú uvažujeme v tvare (Kuropatenková viskozita):

$$q^{Kur}_c = \rho_c \left(C_2 \frac{\gamma + 1}{2} |\Delta w| + \sqrt{C_2^2 \left(\frac{\gamma + 1}{4} \right)^2 (\Delta w)^2 + C_1^2 (c_c)^2} \right) |\Delta w| , \tag{52}$$

kde C_1,C_2 sú konštantý väčšinou rovné 1 a c_c je rýchlosť zvuku v bunke.

Keďže viskozitu definujeme len pre bunku, a teda v každej jej subzóne je rovnaká, bude mať viskózna sila po následnej úprave tvar

$$F^{q_i}_{qn} = +(-1)^i \sum_{l,k=0}^{3,1} q^{l+2}_{c(n,l)} I^{q_i}_{s(n,2l+k-1)} = +(-1)^i \sum_{l,k=0}^{3,1} q^{l+2}_{c(n,l)} (-1)^k I^{q_i}_{a(n,l-k)} \tag{53}$$

Celková sila pôsobiaca na uzol je teda

$$F_n = F_{pn} + F_{qn}$$

Po nahradení časovej derivácie v rovnici pre zákon zachovania hybnosti centrálnou diferenciou potom máme

$$F_n{=}m_n \left(\frac{\partial w}{\partial t}\right)_n = m_n \left(\frac{w_n^{k+1} - w_n^k}{\Delta t}\right) \tag{54}$$

Odkiaľ získavame výsledný vzťah pre rýchlosť pohybu uzlov

$$w_{n^{k+1}} = w_n^k + \frac{\Delta t}{m_n} F_n$$

2.4 Vnútorná energia

Pre odvodenie vzťahu pre vnutornu energiu vychádzame z rovnice pre zákon zachovania energie. Definujme, celkovú energiu v bunke ako

$$E_c = m_c \epsilon_c + \sum_{l=1}^{4} \frac{1}{2} m_c^l \left(\left(w_{n(c,l)}^x\right)^2 + \left(w_{n(c,l)}^y\right)^2 \right)$$

Keďže zákon zachovania energie platí v každej bunke, je časová derivácia celkovej energie v bunke rovná 0. Môžeme potom napísať

$$0 = \frac{\partial E_c}{\partial t} = m_c \frac{\partial \epsilon_c}{\partial t} + \sum_{l=1}^{4} \frac{1}{2} m_c^l \left(2 w_{n(c,l)}^x \frac{\partial w_{n(c,l)}^x}{\partial t} + 2 w_{n(c,l)}^y \frac{\partial w_{n(c,l)}^y}{\partial t} \right)$$

čo spolu s (54) nám dáva

$$0 = m_c \frac{\partial \epsilon_c}{\partial t} + \sum_{l=1}^{4} \left(w_{n(c,l)}^x F_c^{xl} + w_{n(c,l)}^y F_c^{yl} \right)$$

Definujme teraz celkovú pracu vykonanú v bunke silami F_c vzťahom

$$E_c^{work} = -\sum_{l=1}^{4} w_{n(c,l)} F_c^l$$

Potom máme

$$m_c \frac{\partial \epsilon_c}{\partial t} = E_c^{work}$$

Nahradením časovej derivácie centrálnou diferenciou získavame výsledný vzťah pre novú špecifickú vnútornú energiu

$$\epsilon_c^{k+1} = \epsilon_c^k + \frac{\Delta t}{m_c} E_c^{work}$$

2.5 Kompletný krok Lagrangeovej metódy

Na začiatku jednoho Lagrangeovho kroku poznáme nasledujúce veličiny buď z predchádzajúceho kroku alebo z počiatočnej inicializácie: časový krok Δt, rýchlosti uzlov w_n, bunkové a subzonálne objemy V_c, V_c^l, bunkové a subzonálne hustoty ρ_c, ρ_c^l, tlak v bunke p_c, vnútornú energiu bunky ϵ_c. Potom celý krok sa dá popísať nasledujúcim spôsobom

1. Pre každú subzónu spočítame tlakové a viskózne sily $F_{p_c^l}$, $F_{q_c^l}$.

2. Spočítame celkové sile pôsobiace na subzóny a celkové sily pôsobiace na uzly $F_c^l = F_{p_c^l} + F_{q_c^l}$, $F_n = \sum_{l=1}^{4} F_{c(n,l)}^l$.

3. Vzhľadom k získaným silám určíme nové rýchlosti uzlov $w_{n^{k+1}} = w_n^k + \frac{\Delta t}{m_n} F_n$, aplikujeme na ne okrajové podmienky a určíme rýchlosti v polovičnom čase $w_n^{k+1/2} = \frac{1}{2}\left(w_n^k + w_n^{k+1}\right)$.

4. Posunieme uzly na ich nové polohy $z_n^{k+1} = z_n^k + \Delta t w_n^{k+1/2}$ a prepočítame celú geometriu sieťky (stredy buniek, polohy hrán, separátorov, subhrán, bunkové a subzonálne objemy).

5. Určíme celkovú prácu v bunke vykonanú sílami pôsobiacimi na jej uzly $E_c^{work} = -\sum_{l=1}^{4} w_{n(c,l)} F_c^l$ a spočítame novú vnútornú energiu bunky $\epsilon_c^{k+1} = \epsilon_c^k + \frac{\Delta t}{m_c} E_c^{work}$.

6. Dopočítame nové bunkové a subzonálne hustoty $\rho_c = m_c / V_c$, $\rho_c^l = m_c^l / V_c^l$.

7. Ďalej dopočítame nové tlaky podľa stavovej rovnice a aplikujeme tlakovú okrajovú podmienku.

8. Dopočítame ostatné stavové veličiny.

9. Priradíme novej sieťke spočítané nové rýchlosti $w_n = w_n^{k+1}$

Popísaný postup využíva Eulerovu metódu diskretizácie v čase, ktorá je jednoduchá, rýchla avšak s presnosťou prvého rádu. Zlepšenie sa dá dosiahnúť použitím presnejšej metódy napríklad Runge-Kutovej metódy druhého rádu, alebo niektorou metótou prediktor-korektor.

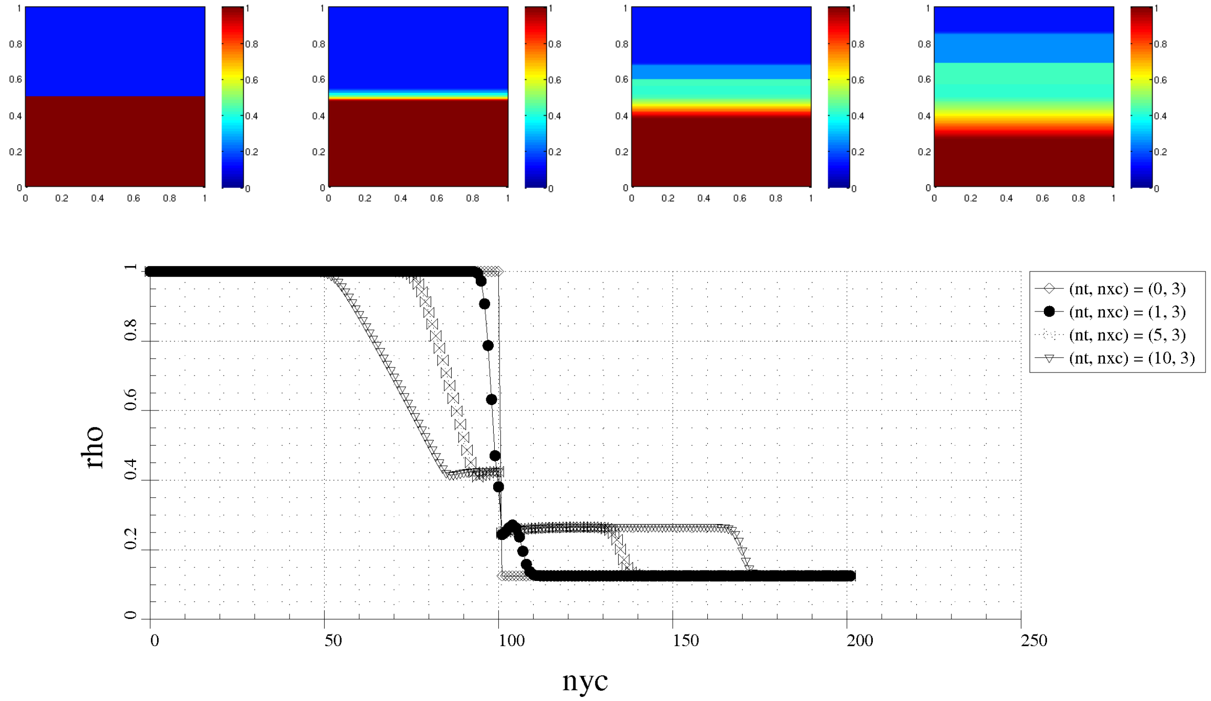
3 Výsledky

Korektnosť nášho algoritmu sme overovali na viacerých úlohách, pre ktoré je známe ich riešenie. Ak nebude napísané inak, na nasledujúcich obrázkoch bude vždy zobrazená hustota materiálu s príslušnou legendou, kde studenšie (modrejšie) miesta budu znamenať oblasť s menšou hustotou než teplejšie miesta (červenejšie) s vyššou hustotou. Podobne, ak nebude napísane inak, uvažujeme ekvidištantné rozdelenie sieťky a hmotnosť každej bunky $m_c = 1$. Venujme sa teda ďalej jednotlivým problémom.

3.1 Sodov problém

Jedným z prvých testovacích problémov bol takzvaný Sodov problém. Jedná sa o Riemanov problém, kde uvažujeme oblasť rozdelenú na dve podoblasti, či už horizontálne alebo vertikálne. V jednej podoblasti s nachádza materiál s menšou hustotou a menším tlakom, v druhej podoblasti je materiál s výššou hustotou a vyšším tlakom. Zvolili sme vertikálne rozdelenie, a to z dôvodu porovnania kartézskej geometrie a cylindrickej geometrie.

Na obrázku (2) môžme vidieť priebeh riešenia ako aj graf závislosti hustoty od polohy na ose y. Výsledky jak v karteziánskej geometriji tak aj v cylindrickej geometrii boli relatívne podobné, a tak na obrázkoch vidíme len riešenie v karteziánskej geometrii.

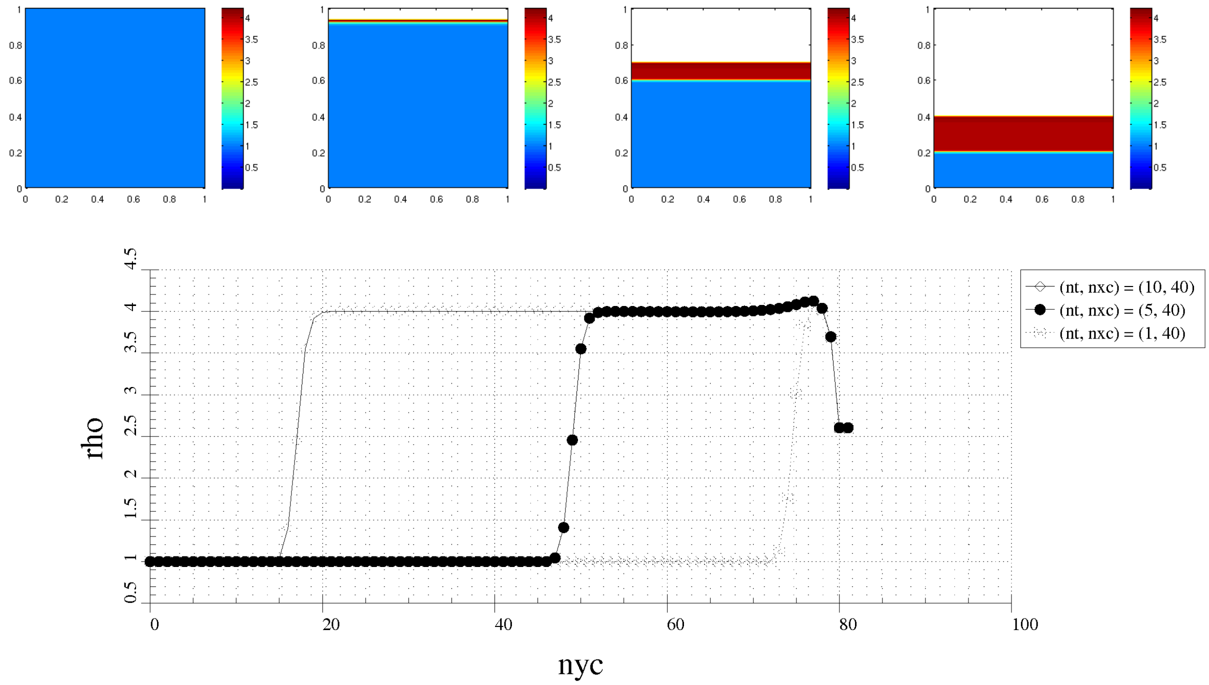


Obrázok 2: Riešenie Sodovho problému s $p_1 = 1$, $\rho_1 = 1$, $p_2 = 0,125$, $\rho_2 = 0,1$. Na prvých 4 obrázkoch môžme vidieť riešenie postupne v časových hladinách $t_0 = 0$, $t_1 = 0,02$, $t_5 = 0,1$ a $t_{10} = 0,2$. Na poslednom obrázku graf závislosti hustoty od polohy na ose y.

3.2 Problém pôsobiaceho piestu

Uvažujme ďalej piest pôsobiaci na materiál. Nech teda hustota a tlak sú v celej doméne rovnaké. Odlišnosť nastáva v rýchlostiach, kde okrajovým uzlom charakterizujúcim piest udelíme nenulovú počiatočnu rýchlosť orientovanú súbežne pre každý uzol smerom do domény. Inými slovami, ak piest pôsobý zhora na doménu, udelíme okrajovým uzlom rovnakú vertikálnu rýchlosť smerom nadol.

Priebeh riešenia spolu s grafom závislosti hustoty na polohe na ose y vidíme na obrázku (3). Opätovne boli výsledky v karteziánskej a cylindrickej geometrii relatívne podobné, a tak uvádzame iba riešenie v karteziánskej geometrii.



Obrázok 3: Riešenie problému pôsobiaceho piestu. Piest pôsobí v zvislom smere s vektorom rýchlosti $w = (0, -1)$. Hustota bola v každej bunke $\rho = 1$, tlak $p = 10^{-6}$. Na prvých 4 obrázkoch vidíme priebeh riešenia postupne v časových hladinách $t_0 = 0$, $t_1 = 0,06$, $t_5 = 0,36$, $t_{10} = 0,6$. Na poslednom obrázku graf závislosti hustoty od polohy na ose y.

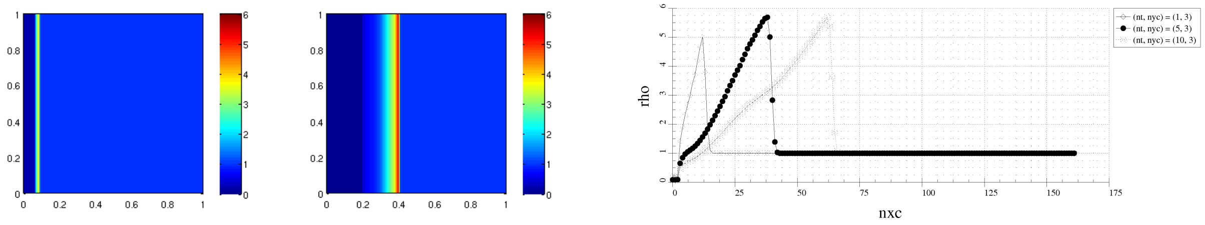
3.3 Sedov problém

V tomto prípade sú opäť hustota a tlak v celej doméne rovnaké až na jednu bunku, tzv. „peak" (a v jednorozmernom karteziánskom resp. dvojrozmernom cylindrickom probléme sme pochopiteľne nastavili celý pás buniek), väčšinou umiestnenú v strede domény, ale z dôvodov symetrie problému môžme umiestniť takúto bunku aj do rohu a spresniť tak riešenie, v ktorej nastavíme počiatočnu energiu relatívne vysokú a podobne aj tlak bude relatívne vyšši než v okolí. To spôsobí tzv. rázovú vlnu, ktorá sa bude šíriť od stredu symetricky smerom k okrajom.

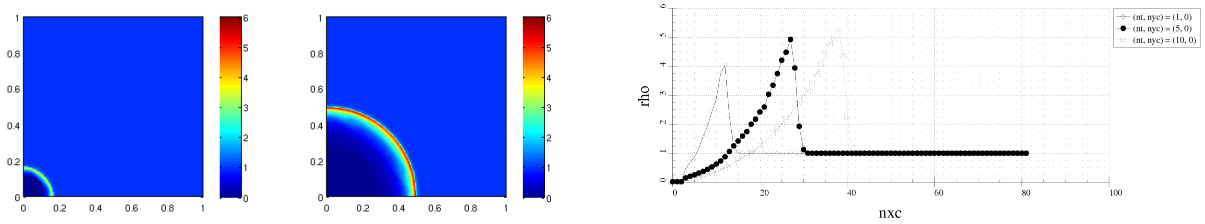
Vhodným nastavením počiatočnej podmienky sme dokázali simulovať 1D Sedov problém v karteziánskej geometrii (viď. obrázok 4), 2D Sedov problém v karteziánskej geometrii (obrázok 5), 2D Sedov problém v cylindrickej geometrii (6) a 3D Sedov problém v cylindrickej geometrii (7). Počiatočná hustota v celej oblasti bola $\rho_{ambient} = \rho_{peak} = 1$, tlak v okolí $p_{ambient} = 10^{-6}$, a v tzv. peaku sme definovali celkovú energiu $E = 0.244816$.

3.4 Nohov problém

Nohov problem je definovaný opäť pre doménu, kde je tlak a hustota všade rovnaká, pričom každému uzlu udelíme rýchlosť rovnakej velkosti orientovanu vždy do jednoho a toho istého bodu (tzv. black hole point). Normu rýchlosti sme volili $|w| = 1$, počiatočnú hustotu $\rho = 1$, tlak $p = 10^{-6}$. Priebehy jednotlivých riešení splolu s grafmi závislostí

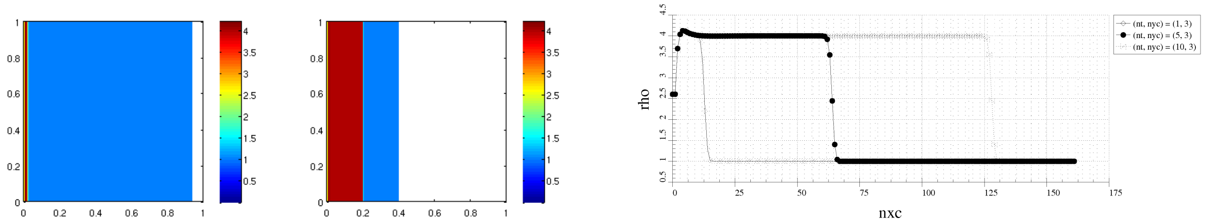


Obrázok 4: Riešenie 1D Sedovho problému v karteziánskej geometrii.

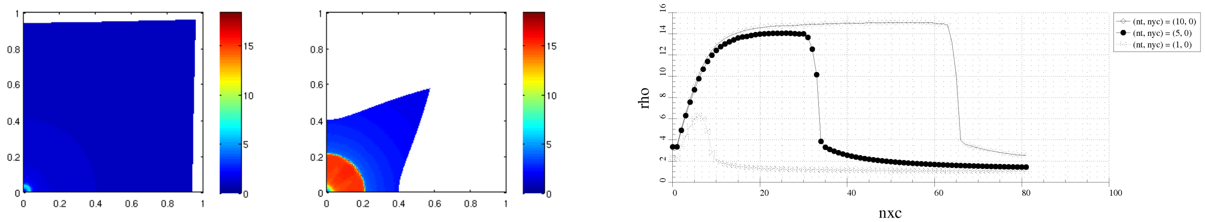


Obrázok 5: Riešenie 2D Sedovho problému v karteziánskej geometrii.

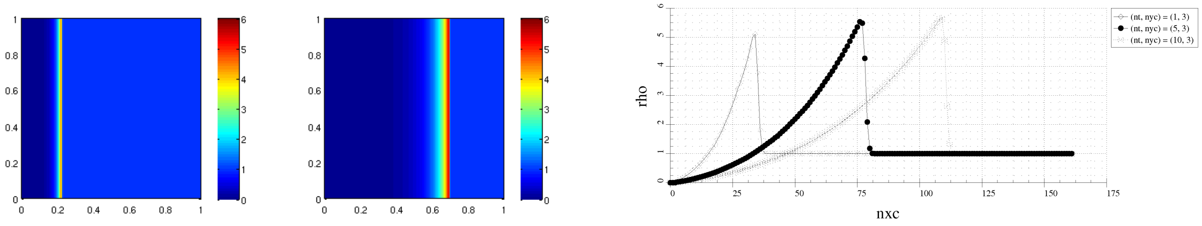
hustoty na polohe na osi x sú zobrazené postupne na obrázkoch (8), (9), (10) a (11).



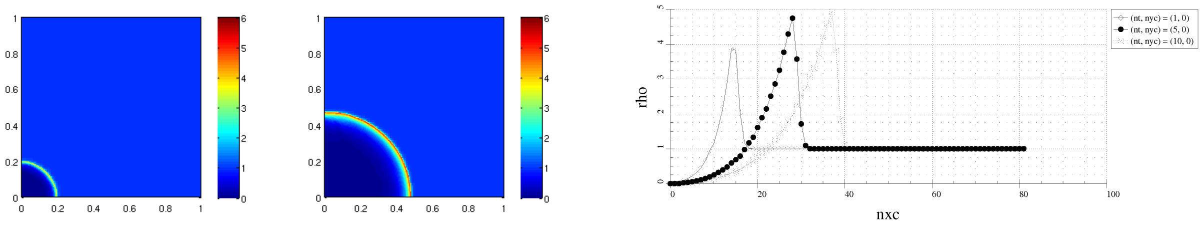
Obrázok 8: Riešenie 1D Nohovho problému v karteziánskej geometrii.



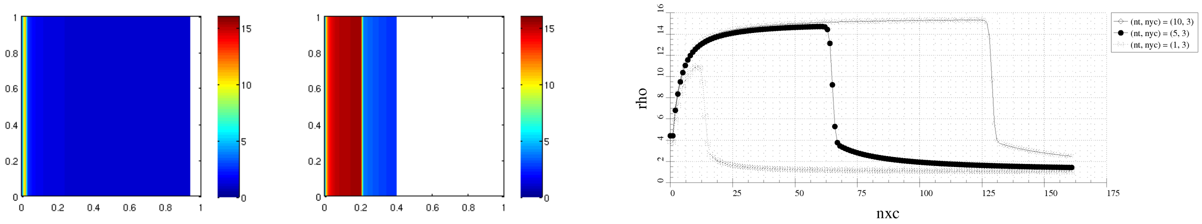
Obrázok 9: Riešenie 2D Nohovho problému v karteziánskej geometrii.



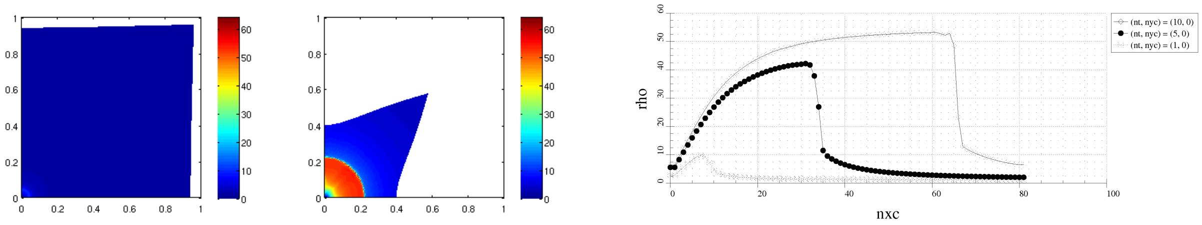
Obrázok 6: Riešenie 2D Sedovho problému v cylindrickej geometrii.



Obrázok 7: Riešenie 3D Sedovho problému v cylindrickej geometrii.



Obrázok 10: Riešenie 2D Nohovho problému v cylindrickej geometrii.



Obrázok 11: Riešenie 3D Nohovho problému v cylindrickej geometrii.

4 Záver

Zoznámili sme sa s modelom Lagrangeovských rovníc pre riešenie hydrodynamiky laserovej plazmy. Ďalej sme sme si ukázali diskretizáciu nášho modelu, a to jak v karteziánskej geometrii tak aj v cylindrickej, pričom sme sa zamerali na získanie kompaktných vzorcov pre oba typy geometrie. Popis nášho algoritmu sme zhrnuli popísaním celého jednoho

Lagrangeovského kroku. Na základe získaných výsledkov zo simulácii vybraných problémov sme overili korektnosť naších výpočtov.

Literatúra

[1] E.J. Caramana, D.E. Burton, M.J. Shashkov., P.P. Whalen. *The Construction of Compatible Hydrodynamics Algorithms Utilizing Conservation of Total Energy*, J. of Com. Phys. (1998), **146**: 227-262.

[2] T. Kapin, M. Kuchařík, J. Limpouch, R. Liska, P. Váchal. *Arbitrary Lagrangian Eulerian method for laser plasma simulations.* Int. J. Numer. Meth. Fluids (2008), **56**: 1337-1342.

[3] T. Kapin, M. Kuchařík, J. Limpouch, R. Liska. *Hydrodynamic simulations of laser interactions with low-density foams.* Czechoslovak Journal of Physics (2006), **56**: B493-B499.

[4] R. Liska, M. Kuchařík. *Arbitrary Lagrangian Eulerian Method for Compressible Plasma Simulations*, Proceedings of Equadiff-11, (2005), pp. 1-10.

[5] P. Havlík. *Diferenční schémata pro hydrodynamiku na nerovnoměrých a Lagrangeovských sítích.* Výzkumní úkol, České vysoké učení technicke v Praze, (2004).

[6] M. Shashkov. *Conservative Finite-Difference Methods on General Grids.* CRC Press, Boca Raton, (1996).

[7] M. Shashkov, B. Wendroff. *A Composite Scheme for Gas Dynamics in Lagrangian Coordinates*, J. of Comp. Phys. (1999), **150**: 502-517.

Transport of Colloids Through Porous Media

Pavel Beneš

2nd year of PGS, email: benespa1@fjfi.cvut.cz
Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, CTU in Prague
advisor: Jiří Mikyška, Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, CTU in Prague

Abstract. The goal of this contribution is to describe the transport of colloids in porous media. This work includes equations describing the flow field, transport of colloids, and deposition of colloids in porous media. Then we describe a numerical discretization of the system of equations describing the colloid transport with known flow field by means of the upwind scheme. We present some numerical results at the end of the contribution.

Abstrakt. Hlavním cílem tohto příspěvku je popis transportu koloidů v porézním prostředí. Tato práce obsahuje rovnice popisující proudové pole, transport koloidů a jejich ukládání v porézním prostředí. Dále je v práci obsažena numerická diskretizace tohoto systému rovnic popisujícího transport koloidů při známém proudovém poli za použití upwindového schématu. Na závěr příspěvku jsou uvedeny některé dosažené výsledky.

1 Introduction

Colloids are small particles with at least one dimension smaller than 100 nm. Because of their size, colloid particles are strongly attracted to the pore surfaces. On the other hand, colloids, like nanoiron particles, can be strongly reactive and can be used in remediation of contamined sites. To plan a suitable remediation strategy, one has to understand mechanisms of colloid transport and their deposition in the subsurface. This understanding can be obtained by means of numerical models. This paper contains equations describing colloidal transport in porous media. Then introduce explicit and semi-explicit schemes and present some results of numerical experiments.

2 The Physical Model

This section presents equations describing the colloidal transport in porous media [1].

2.1 Colloid Transport Equation

The colloid transport equation can be derived from the mass balance of colloids over the REV (representative element volume). There are three main mechanisms controlling the colloidal transport: hydrodynamic dispersion, advection and colloid deposition and release. This can be described by the generalized advection dispersion equation, where the unknown is the particle number concentration n

$$\frac{\partial n}{\partial t} = \nabla \cdot (D\nabla n) - \nabla \cdot (\mathbf{V} \cdot n) - \frac{f}{\pi a_p^2}\frac{\partial \theta}{\partial t}, \tag{1}$$

where θ is the specific surface coverage, defined as

$$\theta = \frac{\text{total cross-section area of deposited colloids}}{\text{interstitial surface area of the porous media solid matrix}},$$

f is specific surface area

$$f = \frac{\text{interstitial surface area}}{\text{porous medium pore volume}},$$

a_p is radius of colloidal particles, D is particle hydrodynamic dispersion tensor and $\mathbf{V}$ is the particle velocity vector. It is possible to write the particle hydrodynamic dispersion tensor as

$$D_{ij} = \alpha_T \bar{V}\delta_{ij} + (\alpha_L - \alpha_T)\frac{\bar{V}_i \bar{V}_j}{\bar{V}} + D_d T \delta_{ij},$$

where D_d is the Stokes-Einstein diffusivity, $\bar{V}_i$, $\bar{V}_j$ are components of the interstitial velocity, α_L is the longitudinal dispersivity, α_T is the transverse dispersivity and T is the tortuosity of the porous medium.

2.2 Colloid Deposition and Release

Let λ be the percentage part of the solid matrix with favorable conditions for colloid deposition. This can be for example areas with iron oxides on its surface. These surfaces are typically positively charged and colloids are typically negatively charged. Deposition on the surfaces is usually irreversible. On the rest $(1-\lambda)$ of the solid matrix surface are unfavorable conditions for the colloidal deposition. Deposition takes place on both parts, but difference in rates can be huge. For particle surface coverage rate we can adopt this patchwise model

$$\frac{\partial \theta}{\partial t} = \lambda \frac{\partial \theta_f}{\partial t} + (1-\lambda)\frac{\partial \theta_u}{\partial t}, \tag{2}$$

where θ_f is favorable surface fraction and θ_u is unfavorable surface fraction. These rates are described by the following partial differential equations

$$\frac{\partial \theta_f}{\partial t} = \pi a_p^2 k_{dep,f} n B(\theta_f) - k_{det,f}\theta_f R(\theta_f), \tag{3}$$

$$\frac{\partial \theta_u}{\partial t} = \pi a_p^2 k_{dep,u} n B(\theta_u) - k_{det,u}\theta_u R(\theta_u), \tag{4}$$

where k_{dep} is the colloid deposition rate constant, k_{det} is the colloid release rate constant, $B(\theta)$ is the dynamic blocking function and $R(\theta)$ is the dynamic release function. The colloid deposition rate coefficient k_{dep} can be expressed by means of a single collector efficiency η

$$k_{dep} = \frac{\eta \varepsilon V}{4} = \frac{\alpha \eta_0 \varepsilon V}{4}, \tag{5}$$

where V is the fluid advection velocity, ε is porosity and η_0 is the favorable single collector removal efficiency.

2.3 Dynamic Blocking and Release Functions $B(\theta)$, $R(\theta)$

Dynamic blocking functions characterize the particle deposition [4]. When the collector is particle free at the beginning, blocking function has value $B(\theta) = 1$. As the deposited particles block the surface more and more, $B(\theta)$ decreases. At the maximum attainable surface coverage $\theta = \theta_{max}$ (jamming limit), $B(\theta) = 0$.

2.3.1 RSA Dynamic Blocking Function

For colloidal particles depositing on the oppositely charged collector surface, these conditions for use of RSA model are valid:

- attachment is irreversible as long as conditions do not change
- surface diffusion is negligible
- particle-particle contact is prohibited

For low and moderate surface coverage the function $B(\theta)$ has this form

$$B(\theta) = 1 - 4\theta_\infty \frac{\theta}{\theta_{max}} + \frac{6\sqrt{3}}{\pi}\left(\theta_\infty \frac{\theta}{\theta_{max}}\right)^2 + \left(\frac{40}{\sqrt{3}\pi} - \frac{176}{3\pi^2}\right)\left(\theta_\infty \frac{\theta}{\theta_{max}}\right)^3,$$

where θ_∞ is the hard sphere jamming limit.

2.3.2 Dynamic Release Function

The dynamic release function describes the probability of colloid release from the porous media surface covered by retained colloids [1]. This function should in general depend on the colloid residence time and the retained colloid concentration. Because the colloid release is not well understood, we will use $R(\theta) = 1$.

3 Mathematical Model

This section shows solved equations, initial and boundary conditions. By substituting equations describing the colloid deposition and release (2), (3) and (4) into (1), we obtain the following expression

$$\frac{\partial n}{\partial t} = \nabla \cdot (D\nabla n) - \nabla \cdot (\mathbf{V} \cdot n) - \frac{f}{\pi a_p^2}((\lambda \pi a_p^2 k_{dep,f} B(\theta_f) + (1-\lambda)\pi a_p^2 k_{dep,u} B(\theta_u))n - ((\lambda \pi k_{det,f} \theta_f R(\theta_f) + (1-\lambda) k_{dep,u} \theta_u R(\theta_u)). \quad (6)$$

We assume that $K(\theta) = 1$ (first-order kinetics release mechanism) and use the following notations

$$\begin{aligned} \gamma &= \tfrac{f}{\pi a_p^2}, \\ K_a(\theta_f, \theta_u) &= \pi a_p^2[\lambda k_{dep,f} B(\theta_f) + (1-\lambda) k_{dep,u} B(\theta_u)], \\ K_r(\theta_f, \theta_u) &= \lambda \pi k_{det,f} \theta_f + (1-\lambda) k_{dep,u} \theta_u. \end{aligned} \quad (7)$$

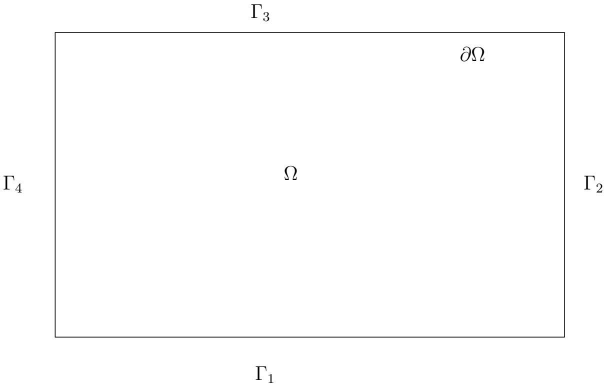


Figure 1: The domain Ω.

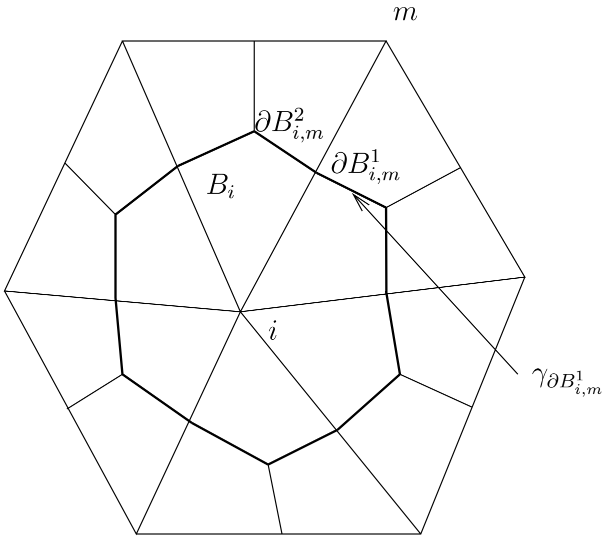


Figure 2: The exclusive subdomain for node i.

Under these assumptions, the following equation is obtained

$$\frac{\partial n}{\partial t} = \nabla \cdot (D\nabla n) - \nabla \cdot (\mathbf{V} \cdot n) - \frac{K_a(\theta_f, \theta_u)}{\gamma} n + \frac{K_r(\theta_f, \theta_u)}{\gamma}. \tag{8}$$

In (8), V is a known velocity field given by a flow model. We complete this equation with (3) and (4)

$$\begin{aligned}\frac{\partial \theta_f}{\partial t} &= \pi a_p^2 k_{dep,f} n B(\theta_f) - k_{det,f}\theta_f, & (9)\\ \frac{\partial \theta_u}{\partial t} &= \pi a_p^2 k_{dep,u} n B(\theta_u) - k_{det,u}\theta_u. & (10)\end{aligned}$$

To solve this system, we will need boundary conditions for equation (8) and initial conditions for each equation (8), (9) and (10). Let us have rectangular domain Ω with boundary Γ, where lower boundary is denoted Γ_1, right Γ_2, upper Γ_3 and left Γ_4 (Fig. 1).

For concentration equation (8) will have an initial condition

$$n(\mathbf{x}, 0) = n_0(\mathbf{x}) \text{ for } \mathbf{x} \in \Omega, \tag{11}$$

and boundary conditions describing concentration of colloids on Γ

$$n(\mathbf{x}, t) = n_i(\mathbf{x}, t) \text{ for } \mathbf{x} \in \Gamma_i,\ i \in 1, \dots, 4\ . \tag{12}$$

For equations (9) and (10) we need to prescribe initial conditions for θ_f and θ_u. As there are initially no deposited colloids,

$$\theta_f(\mathbf{x}, 0) = \theta_u(\mathbf{x}, 0) = 0 \text{ for } \mathbf{x} \in \Omega. \tag{13}$$

4 Numerical Solution

We discuss the discretization methods for solving (8), (9) and (10). Although our numerical solution is computed on a rectangular grid, we develop the scheme for a more general case of an unstructured mesh in two dimensions composed of triangles and quadrangles of the domain Ω, which is called the primary mesh. We construct a dual mesh by connecting barycentres of each element with midpoints of all its sides in each element from the primary grid. In this way we obtain a polygon around each node from the primary mesh (on the boundary of the domain $\partial\Omega$, polygons are incomplete). For a primary mesh node i, we call this polygon B_i, the exclusive subdomain of node i. ∂B_i consists of several abscissae and each of abscissa belongs to one abscissa connecting node i with his neighbor m. For each couple i, m, there are two abscissae, we denote them $\partial B^l_{i,m}$. The midpoint of the abscissa $\partial B^l_{i,m}$ is denoted $\gamma_{\partial B^l_{i,m}}$ (Fig. 2). The time level is denoted by superscript k. The length of abcissa $\partial B^l_{i,m}$ is denoted $|\partial B^l_{i,m}|$.

4.1 Explicit Scheme

Equation (8) is discretizated using the finite volume method. First we integrate this equation over an element B_i

$$\int_{B_i}\left[\frac{\partial n}{\partial t}+\frac{K_a(\theta_f,\theta_u)}{\gamma}n-\frac{K_r(\theta_f,\theta_u)}{\gamma}\right]\mathrm{d}S=\int_{B_i}\left[\nabla\cdot(D\nabla n)-\nabla\cdot(\mathbf{V}\cdot n)\right]\mathrm{d}S. \tag{14}$$

Now we use Gauss formula on the right hand side of equation (14)

$$\int_{B_i}\left[\nabla\cdot(D\nabla n)-\nabla\cdot(\mathbf{V}\cdot n)\right]\mathrm{d}S=\int_{\partial B_i}(D\nabla n)\cdot\mathbf{n}_{\partial B_i}\mathrm{d}l-\int_{\partial B_i}(\mathbf{V}\cdot n)\cdot\mathbf{n}_{\partial B_i}\mathrm{d}l, \tag{15}$$

where ∂B_i is boundary of B_i and $\mathbf{n}_{\partial B_i}$ is the normal vector to ∂B_i.

Equations (14) and (15) give together

$$\int_{B_i}\left[\frac{\partial n}{\partial t}+\frac{K_a(\theta_f,\theta_u)}{\gamma}n-\frac{K_r(\theta_f,\theta_u)}{\gamma}\right]\mathrm{d}S=\int_{\partial B_i}(D\nabla n)\cdot\mathbf{n}_{\partial B_i}\mathrm{d}l-\int_{\partial B_i}(\mathbf{V}\cdot n)\cdot\mathbf{n}_{\partial B_i}\mathrm{d}l. \tag{16}$$

The left hand side of (16) is approximated as

$$\begin{aligned}\int_{B_i}\left[\frac{\partial n}{\partial t}+\frac{K_a(\theta_f,\theta_u)}{\gamma}n-\frac{K_r(\theta_f,\theta_u)}{\gamma}\right]\mathrm{d}S\approx&\\ \left[\frac{n_i^{k+1}-n_i^k}{\Delta t}+\frac{K_a(\theta^k_{f,i},\theta^k_{u,i})}{\gamma}n_i^k-\frac{K_r(\theta^k_{f,i},\theta^k_{u,i})}{\gamma}\right]|B_i|.&\end{aligned} \tag{17}$$

The first term on the right hand side of (16) can be approximated as

$$\begin{aligned}\int_{\partial B_i}(D\nabla n)\cdot\mathbf{n}_{\partial B_i}\mathrm{d}l &= \sum_{m,l}\int_{\partial B^l_{i,m}}(D\nabla n)\cdot\mathbf{n}_{\partial B^l_{i,m}}\mathrm{d}l\\ &\approx \sum_{m,l}\left[(D(\gamma_{\partial B^l_{i,m}})(\nabla n)^k(\gamma_{\partial B^l_{i,m}}))\cdot\mathbf{n}_{\partial B^l_{i,m}}|\partial B^l_{i,m}|\right],\end{aligned} \tag{18}$$

where $(\nabla n)_i^k$ is the approximation of ∇n from concentration values from time level k. The second term on the right hand side of (16) is approximated as

$$\int_{\partial B_i} (\mathbf{V} \cdot n) \cdot \mathbf{n}_{\partial B_i} \mathrm{d}l = \sum_{m,l} \int_{\partial B^l_{i,m}} (\mathbf{V}(\gamma_{\partial B^l_{i,m}}) \cdot n^\star_{i,m,l}) \cdot \mathbf{n}_{\partial B^l_{i,m}} |\partial B^l_{i,m}|, \tag{19}$$

where upwind value reads as

$$n^\star_{i,m,l} = \begin{cases} n_i^k & \text{for } \mathbf{n}_{\partial B^l_{i,m}} \cdot \mathbf{V}(\gamma_{\partial B^l_{i,m}}) > 0, \\ n_m^k & \text{for } \mathbf{n}_{\partial B^l_{i,m}} \cdot \mathbf{V}(\gamma_{\partial B^l_{i,m}}) \leq 0. \end{cases} \tag{20}$$

The approximation (19) is called the first-order upwind scheme and helps us to avoid oscillations in the solution, but suffers from the numerical diffusion. To obtain smaller numerical diffusion without oscillations, higher-order upwind scheme with limiters can be used.

Values of n_i^k on the boundary $\partial\Omega$ are taken from the boundary conditions (11). θ_f and θ_u for the first time step can be θ_f and θ_u taken from the initial condition (13). The explicit scheme for solving (8) is

$$\begin{aligned} n_i^{k+1} = n_i^k - \frac{\Delta t}{|B_i|} \Bigg[|B_i| \left(\frac{K_a(\theta^k_{f,i}, \theta^k_{u,i})}{\gamma} n_i^k - \frac{K_r(\theta^k_{f,i}, \theta^k_{u,i})}{\gamma} \right) + \\ - \sum_{m,l} \left[(D(\gamma_{\partial B^l_{i,m}})(\nabla n)^k(\gamma_{\partial B^l_{i,m}})) \cdot \mathbf{n}_{\partial B^l_{i,m}} |\partial B^l_{i,m}| \right] + \\ \sum_{m,l} \int_{\partial B^l_{i,m}} (\mathbf{V}(\gamma_{\partial B^l_{i,m}}) \cdot n^\star_{i,m,l}) \cdot \mathbf{n}_{\partial B^l_{i,m}} |\partial B^l_{i,m}| \Bigg]. \end{aligned} \tag{21}$$

We use the forward Euler scheme to discretize equations (9) and (10) to compute particle coverage: θ_l^{k+1} for favorable case $l = f$ and unfavorable case $l = u$

$$\theta^{k+1}_{l,i} = \theta^k_{l,i} + \Delta t \left(\pi a_p k_{dep,l,i} n_i^{k+1} B(\theta^k_{l,i}) - k_{det,l,i} \theta^k_{l,i} \right) \qquad l \in \{\text{f,u}\}, \tag{22}$$

where $\theta^k_{l,i}$, $k_{dep,l,i}$, $k_{det,l,i}$ are values of θ^k_l, $k_{dep,l}$, $k_{det,l}$ in the node i. In equation (22), we use n_i^{k+1} that has been computed previously by (21).

The coupled system of equations is solved as follows. We denote by Δt the time step.

- 1. Initial conditions are initialized, $k = 1$
- 2. The number concentration n^k, based on the number concentration and coverage at time level $k - 1$, in time $t_k = k\Delta$t is computed by the scheme (21)
- 3. New surface coverages θ_f^k and θ_f^k are computed from (22) using values of n^k obtained from 2. and surface coverages from time level $k - 1$
- 4. If $k\Delta t <$ (Final time) increase k by 1 and go to 2; else end.

4.2 Semi-implicit Numerical Scheme

Explicit scheme has the disadvantage that the time steps has to be limited due to CFL condition [6]. For this reason we implemented semi-implicit numerical scheme [5], which will enable us to use larger time steps compared to the explicit scheme.

Equation (8)

$$\frac{\partial n}{\partial t} = \nabla \cdot (D \nabla n) - \nabla \cdot (\mathbf{V} \cdot n) - \frac{K_a(\theta_f, \theta_u)}{\gamma} n + \frac{K_r(\theta_f, \theta_u)}{\gamma}. \tag{23}$$

is solved using the operator splitting technique. At first we solve explicitly convection and reaction parts of the equation

$$\frac{\partial n}{\partial t} = -\nabla \cdot (\mathbf{V} \cdot n) - \frac{K_a(\theta_f, \theta_u)}{\gamma} n + \frac{K_r(\theta_f, \theta_u)}{\gamma} \tag{24}$$

obtained from (8) by setting $D = 0$. We discretize (23) as follows

$$\left[\frac{n_i^{k+\frac{1}{2}} - n_i^k}{\Delta t} + \frac{K_a(\theta_{f,i}^k, \theta_{u,i}^k)}{\gamma} n_i^k - \frac{K_r(\theta_{f,i}^k, \theta_{u,i}^k)}{\gamma}\right] |B_i| + \sum_{m,l} \int_{\partial B_{i,m}^l} (\mathbf{V}(\gamma_{\partial B_{i,m}^l}) \cdot n_{i,m,l}^\star) \cdot \mathbf{n}_{\partial B_{i,m}^l} |\partial B_{i,m}^l| = 0, \tag{25}$$

where the upwind value is given as (20). The value of $n_i^{k+\frac{1}{2}}$ is used as an initial condition and (12) as boundary conditions for solving the diffusion equation

$$\frac{\partial n}{\partial t} = \nabla \cdot (D \nabla n), \tag{26}$$

which is solved using the backward Euler scheme

$$\left[\frac{n_i^{k+1} - n_i^{k+\frac{1}{2}}}{\Delta t}\right] |B_i| = \sum_{m,l} \left[(D(\gamma_{\partial B_{i,m}^l})(\nabla n)^{k+1}(\gamma_{\partial B_{i,m}^l})) \cdot \mathbf{n}_{\partial B_{i,m}^l} |\partial B_{i,m}^l|\right]. \tag{27}$$

We denote number of nodes in one row of our numerical grid n_r. On a rectangular grid with grid sizes Δx, Δy, equation (27) reads as

$$\left[\frac{n_i^{k+1} - n_i^{k+\frac{1}{2}}}{\Delta t}\right] |B_i| - \Delta x D_{yy}(\gamma_{\partial B_{i,i+n_r}}) \left(\frac{n_{i+n_r}^{k+1} - n_i^{k+1}}{\Delta y}\right) + \Delta y D_{xx}(\gamma_{\partial B_{i,i-1}}) \left(\frac{n_i^{k+1} - n_{i-1}^{k+1}}{\Delta x}\right) + \Delta x D_{yy}(\gamma_{\partial B_{i,i-n_r}}) \left(\frac{n_i^{k+1} - n_{i-n_r}^{k+1}}{\Delta y}\right) - \Delta y D_{xx}(\gamma_{\partial B_{i,i+1}}) \left(\frac{n_{i+1}^{k+1} - n_i^{k+1}}{\Delta x}\right) = 0. \tag{28}$$

In equation (28) the terms containing boundary values can be eliminated into right hand side. In every time step we need to solve the system $An^{k+1} = b$, where

$$
\begin{aligned}
A_{i,i-n_r} &= -\frac{\Delta x D_{yy}(\gamma_{\partial B_{i,i-n_r}})}{\Delta y} \\
A_{i,i-1} &= -\frac{\Delta y D_{xx}(\gamma_{\partial B_{i,i-1}})}{\Delta x} \\
A_{i,i} &= \frac{|B_i|}{\Delta t} + \frac{\Delta x D_{yy}(\gamma_{\partial B_{i,i+n_r}})}{\Delta y} + \frac{\Delta y D_{xx}(\gamma_{\partial B_{i,i-1}})}{\Delta x} + \frac{\Delta x D_{yy}(\gamma_{\partial B_{i,i-n_r}})}{\Delta y} + \frac{\Delta y D_{xx}(\gamma_{\partial B_{i,i+1}})}{\Delta x} \\
A_{i,i+1} &= -\frac{\Delta y D_{xx}(\gamma_{\partial B_{i,i+1}})}{\Delta x} \\
A_{i,i+n_r} &= -\frac{\Delta x D_{yy}(\gamma_{\partial B_{i,i+n_r}})}{\Delta y}
\end{aligned}
$$

and $A_{i,j} = 0$ elsewhere. The right hand side of the solved system b reads as

$$
b_i = \frac{|B_i|}{\Delta t} n_i^k, \tag{29}
$$

where index i goes through all nodes. The boundary terms can be eliminated into the b_i.

5 Results

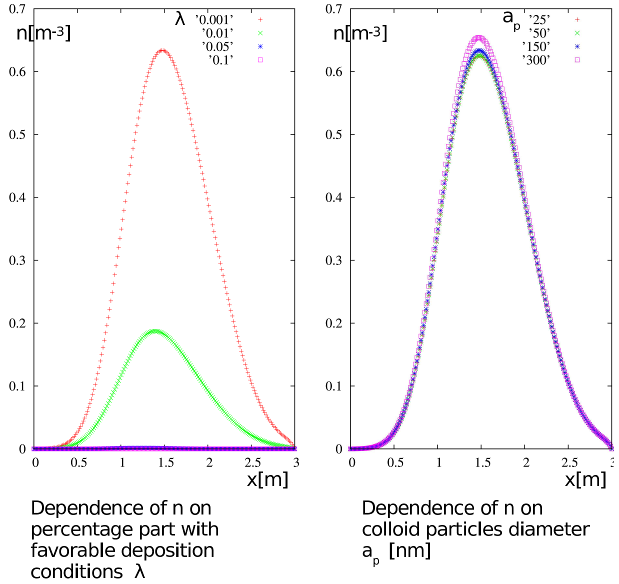
In this section we present results using data set from [1] of a two dimensional experiment. In [1] dependence of solution on physical parameters percentage part with favorable conditions λ and colloid particle diameter a_p was investigated.

We are given a stationary flow field parallel to the x-axis on a square domain Ω of size 3×3m. In the beginning no colloidal particles are present in the area. We prescribe a boundary condition

$$
n(\mathbf{x}, t) = \begin{cases} 2.8 \cdot 10^{14} [m^{-3}] & \text{for} \quad t \leq 0.5\text{day} \\ 0 & \text{for} \quad 1.0 \geq t > 0.5\text{day} \end{cases} \tag{30}
$$

on Γ_4 and $n(\mathbf{x}, t) = 0$ for $x \in \Gamma_1,\ \Gamma_2,\ \Gamma_3$ and $t \in [0, 1\text{day}]$ We are interested in the distribution of colloids in domain Ω in time of one day. Our results are showing the number concentration of colloidal particles contained in water in pores n divided by $2.8 \cdot 10^{14}$, so that the resulting values are rescaled between 0 and 1. The numerical grid with 100×100 nodes was used for computations. Using semi-implicit scheme instead of explicit scheme enabled us to use about six times longer time steps. For explicit scheme we used time step $\frac{1}{420}$ day and for semi-implicit $\frac{1}{70}$ day.

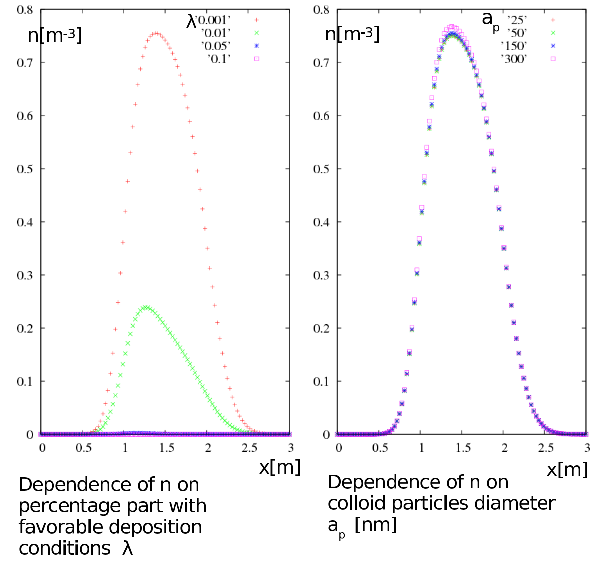
Results computed by the explicit and semi-implicit numerical schemes on a rectangular mesh are shown in Figures 3 and 4. These figures show a cut of the two dimensional solution along the x-axis and show the dependence of number concentration on physical parameters:percentage part with favorable conditions λ and colloid particle diameter a_p. Figures 3 and 4 are depicting that results for both explicit and semi-implicit schemes are close to each other. Figures describing the dependence of the number concentration n on the percentage part with favorable deposition conditions λ show that the concentration of colloids which do not deposit and stay in water decreases quite fast with increasing λ. The concentration of colloids which do not deposit and stay in water decreases with decreasing radius of colloidal particles.



Dependence of n on percentage part with favorable deposition conditions λ

Dependence of n on colloid particles diameter a_p [nm]

Figure 3: Number concentration n divided by 10^{14} for different values of λ and a_p in time 1 day ; explicit numerical scheme.



Dependence of n on percentage part with favorable deposition conditions λ

Dependence of n on colloid particles diameter a_p [nm]

Figure 4: Number concentration n divided by 10^{14} for different values of λ and a_p in time 1 day; semi-implicit scheme.

6 Conclusion

In this contribution a summary of equations describing the colloid transport was presented. The equations were discretized by means of

- the explicit numerical scheme
- the semi-implicit scheme based on the operator splitting technique

both using first order upwind (20) for approximation of the convection term. Numerical results show that by using semi-implicit scheme, we can use approximately six-times longer time steps. Both numerical schemes are in good comparison. We were able to reproduce some of the numerical results from [1]. The dependence of the number concentration of colloids is in good agreement with [1] but there are some discrepancies in the dependence on particle radius (Figures 3 and 4). These discrepancies will be one of subjects of future research. The future work will be focused on a behavior of colloids and nanocolloids in porous media, especially in heterogeneous porous media.

7 Acknowledgment

This work was supported by grants:

- Mathematical Modeling of Multi-Phase Porous Media Flow, project No. 201/08/P567 of the Grant Agency of the Czech Republic, 2008-2010 (principal investigator J. Mikyška)

References

[1] N. Sun, M. Elimelech, N.-Z. Sun *A novel two-dimensional model for colloid transport in physically and geochemically heterogeneous porous media.* Journal of Contaminant Hydrology 49, (2001), 173–199.

[2] N.-Z. Sun *Mathematical Modeling of Groundwater Pollution.* Springer-Verlag, New York.

[3] N.-Z. Sun W.W.-G, Yeh *A proposed upstream weight numerical method for simulating pollutant transport in groundwater.* Water Resour. Res. 19 (1983), 1489–1500.

[4] J.N. Ryan, M. Elimelech *Review Colloid mobilization and transport in groundwater.* Colloids and Surfaces A: Physicochemical and Engineering Aspects 107 (1996), 1–56.

[5] R.J. LeVeque, J. Oliger *Numerical methods based on additive splittings for hyperbolic partial differential equations.* Math. Comp. 40 162 (1983), 469–497.

[6] R.J. LeVeque *Finite-Volume Methods for Hyperbolic Problems.* Cambridge Press, (2002)

Numerical Simulation of Dynamic Capillary Pressure

Radek Fučík

3rd year of PGS, email: fucik@fjfi.cvut.cz
Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, CTU in Prague
advisor: Jiří Mikyška, Department of Mathematics, FNSPE, CTU in Prague

Abstract. In order to investigate effects of the dynamic capillary pressure-saturation relationship used in the modelling of flow in porous medium, a one-dimensional fully implicit numerical scheme is proposed. The numerical scheme is used to simulate experimental procedure using the measured dataset for the sand and fluid properties. Results of the simulation using different models for dynamic effect term in capillary pressure – saturation relationship are presented and discussed.

Abstrakt. V článku je prezentován jednorozměrný model dvoufázového nemísivého a nestlačitelného proudění, který je použit na zkoumání vlivu dynamického efektu pro model kapilárního tlaku v závislosti na saturaci v porézním prostředí. Numerický model je použit k simulaci laboratorních experimentů s cílem posoudit vliv různých modelů pro koeficient dynamického efektu na řešení jednorozměrné úlohy.

1 Introduction

In the description of the behaviour of immiscible and incompressible fluids within porous media, a rigorous definition and a reliable model of the capillarity are crucial. In the past decades, various capillary pressure – saturation models were correlated from laboratory experiments under equilibrium conditions. These *static* capillary pressure models such as [4] or [25] have been used in most of the mathematical studies on modelling of multiphase flow in porous medium. However, it was found out hat the laboratory measured capillary pressure does not correspond to the capillary pressure in the case of large velocities. As a result of the empirical approach in [24], new two-phase flow theories appeared in [12], [13], [15], [14], [7], or [3]. The most important result is that the static capillary pressure – saturation relationship cannot be used in the modelling of capillarity when the fluid content is in motion and, therefore, a new model of the capillary pressure – saturation relationship is proposed and referred to as the dynamic capillary pressure [12], [13], [15], [14].

The two-phase flow system can be simplified to the Richards problem, in which the pressure of the non-wetting phase (air or oil) is assumed to be constant. This is the case in [18], where the dynamic effects are not found to be relevant for the given structure of heterogeneous porous medium. Other numerical approaches using the dynamic capillary pressure have been studied in [20], [19], or [22]. However, the relevance of using the dynamic capillary pressure in the full two-phase flow system of equations has not been fully answered yet. For instance, in [17], the authors present a semi-implicit numerical scheme based on the first-order upwind finite volume method, where the material interfaces are treated by the Lagrange multiplier. However, in that paper, only the constant dynamic effect coefficient was considered whereas other researchers suggest more general functional models for the dynamic effect coefficient as in [23]. A fully implicit numerical scheme is proposed that can be used for a detailed investigation of the saturation and capillary pressure behaviour when dynamic capillary pressure is used instead of the static capillary pressure in the full two-phase flow system. The aim is to investigate behaviour of different functional models of the dynamic capillary pressure coefficient. Moreover, the material interface condition for the dynamic capillary pressure is treated in a new, modified way based on the standard extended capillary pressure condition as in [16].

2 Mathematical model

In this section, we present the mathematical model describing two-phase flow in a one-dimensional porous medium. In this paper, two phases - a wetting phase (indexed w) and a non-wetting phase (indexed n) - are considered to be present within the pores of a porous medium and both fluids are assumed to be incompressible and immiscible. Under these assumptions, the following one-dimensional $p_w - S_n$ formulation in a domain $\Omega = [0, L]$ (see [1]) is given by

$$\Phi \frac{\partial S_\alpha}{\partial t} + \frac{\partial u_\alpha}{\partial x} = 0, \tag{1}$$

$$u_\alpha = -\frac{k_{r\alpha}}{\mu_\alpha} K \left(\frac{\partial}{\partial x}(p_w + \delta_{\alpha n} p_c) - \rho_\alpha \, g \right), \tag{2}$$

where $S_w + S_n = 1$, $\delta_{\alpha n}$ is the Kronecker symbol, and $\alpha \in \{w, n\}$. S_α denotes the saturation, p_α si the pressure, ρ_α is the volumetric density, μ_α is the dynamic viscosity, $k_{r\alpha}$ is the relative permeability of the phase α, where $\alpha \in \{w, n\}$. The Darcy velocities are denoted by u_α. Symbols Φ, K, and g stand for porosity, permeability of the soil matrix and gravitational acceleration, respectively.

Governing equations (1) and (2) are subject to an initial condition

$$S_\alpha = S_\alpha^0, \quad \text{in} \quad \Omega, \tag{3}$$

and boundary conditions

$$u_\alpha \cdot n = u_\alpha^N, \quad \text{on} \quad \Gamma_{u_\alpha}^N, \tag{4}$$

$$S_\alpha = S_\alpha^D, \quad \text{on} \quad \Gamma_S^D, \tag{5}$$

$$p_\alpha = p_\alpha^D, \quad \text{on} \quad \Gamma_{p_\alpha}^D, \tag{6}$$

where n denotes the outer normal vector to the boundary. Generally, $\Gamma_{u_\alpha}^N$ Γ_S^D, and $\Gamma_{p_\alpha}^D$ denote subsets of the boundary Γ of the domain Ω, here, $\Gamma = \{0, L\}$.

Following the standard definitions in literature, the capillary pressure p_c on the pore scale is defined as the difference between the non-wetting phase pressure p_n and the wetting phase pressure p_w, i.e.,

$$p_c = p_n - p_w. \tag{7}$$

On the macroscale, the capillary pressure has been commonly considered as a function of the wetting phase saturation only [16], [2], [11], [8], [9], or [10]. The following Brooks and Corey [4] capillary pressure - effective wetting phase saturation parameterization is used in the presented two-phase flow model [1]

$$p_c^{eq} = p_d (S_w^e)^{-\frac{1}{\lambda}}, \tag{8}$$

where p_d is the entry pressure, λ is the pore size distribution index, and S_w^e is the effective saturation of the wetting phase defined as

$$S_\alpha^e = \frac{S_\alpha - S_{r\alpha}}{1 - \sum\limits_\beta S_{r\beta}}, \tag{9}$$

where $S_{r\alpha}$ is the α-phase irreducible saturation.

The Brooks and Corey relationship (8) is suitable for modelling of flow in heterogeneous porous media because the difference in the entry pressure coefficients p_d in different porous materials captures the barrier effect that has been observed in various experiments [21], [16], [1]. Together with the Brooks and Corey model of p_c given in (8), the Burdine model [5] for the relative permeability functions $k_{r\alpha}$ is given as

$$k_{rw} = (S_w^e)^{3+\frac{2}{\lambda}}, \quad k_{rn} = (1 - S_w^e)^2 \left(1 - (S_w^e)^{1+\frac{2}{\lambda}}\right). \tag{10}$$

The dynamic capillary pressure – saturation relationship is proposed in the following form [13]:

$$p_c := p_n - p_w = p_c^{eq} - \tau \frac{\partial S_w}{\partial t}, \tag{11}$$

where p_c^{eq} is the capillary pressure – saturation relationship in the thermodynamic equilibrium of the system and τ, the dynamic effect coefficient, is a material property of the system.

Early in 1978, before the thermodynamic definition of the capillary pressure (11) in [13], Stauffer [24] observed the dynamic effect in laboratory experiments and proposed a linear dependence in (11) with the following definition of τ

$$\tau_S = \alpha_S \frac{\mu_w \Phi}{K\lambda} \left(\frac{p_d}{\rho_w g}\right)^2, \tag{12}$$

where $\alpha_S = 0.1$ denotes a scaling parameter. Both λ and p_d are the Brooks and Corey parameters [4] that can be experimentally estimated.

The Stauffer model for the dynamic effect coefficient τ_S was obtained by correlating experimental data for fine sands. The values of τ_S vary between $\tau_S = 2.7 \cdot 10^4$ $Pa\ s$ and $\tau_S = 7.7 \cdot 10^4$ $Pa\ s$, see [19, page 27]. Other researchers suggest that the magnitude of τ should be smaller, i.e., in the order of $10^2 - 10^3$ $Pa\ s$ according to [6], or, on the other hand, it should be higher, i.e., in the order of $10^4 - 10^8$ $Pa\ s$ as estimated in [14].

[1] A superscript ^{eq} is used in the definition (8) with respect to the latter and it indicates the model of the capillary pressure for the system in the state of thermodynamic equilibrium.

In this paper, the authors rely on laboratory data, where a more general nonlinear dependence $\tau = \tau(S_w)$ is assumed and the order of magnitude of τ is about 10^6 $Pa\ s$, see Table 3. As a result of the laboratory data, different functional models of $\tau(S_w)$ were correlated and used in the numerical simulations in order to investigate their influence on the two-phase flow. The laboratory experiment is described briefly in Section 4 and in detail in [23].

3 Numerical model

We propose a standard finite volume discretization technique in order to determine approximate discrete solutions $S^k_{n,i}$ and $p^k_{w,i}$ of the problem (1), generally denoted by $f^k_i = f(k\Delta t, i\Delta x)$, where $i = 0, 1, \dots, m$, $m\Delta x = L$, $k = 0, 1, \dots, n$, and $n\Delta t = T$. L denotes the length of the domain and T is the final time of the simulation.

The fully implicit numerical scheme reads

$$\Phi \frac{S^{k+1}_{\alpha,i} - S^k_{\alpha,i}}{\Delta t} = -\frac{u^{k+1}_{\alpha,i+1/2} - u^{k+1}_{\alpha,i-1/2}}{\Delta x}, \tag{13}$$

where $\alpha \in \{w, n\}$. The discrete Darcy velocities u_α introduced by (2) are given by

$$u^{k+1}_{\alpha,i+1/2} = -\frac{K}{\mu_\alpha} k_{r\alpha}(S^{k+1}_{\alpha,upw}) \Big(\underbrace{\frac{p^{k+1}_{w,i+1} - p^{k+1}_{w,i}}{\Delta x} + \delta_{\alpha n} \frac{p^{k+1}_{c,i+1} - p^{k+1}_{c,i}}{\Delta x} - \rho_\alpha\, g}_{\Delta\Phi_\alpha} \Big), \tag{14}$$

and the discrete capillary pressure by

$$p^{k+1}_{c,i} = p_c \left(1 - S^{k+1}_{n,i}, -\frac{S^{k+1}_{n,i} - S^k_{n,i}}{\Delta t} \right) = p^{eq}_c(1 - S^{k+1}_{n,i}) + \tau(1 - S^{k+1}_{n,i}) \frac{S^{k+1}_{n,i} - S^k_{n,i}}{\Delta t}. \tag{15}$$

where $S^{k+1}_{\alpha,upw}$ is the saturation taken in the upstream direction with respect to the gradient of the phase potential Φ_α, i.e.

$$S^{k+1}_{\alpha,upw} = \begin{cases} S^{k+1}_{\alpha,i+1} & \text{if } \Delta\Phi_\alpha \geq 0. \\ S^{k+1}_{\alpha,i} & \text{if } \Delta\Phi_\alpha < 0. \end{cases}$$

The fully implicit numerical scheme is solved using the Newton-Raphson iteration method. The Jacobi matrix is block tridiagonal and therefore solved by the Thomas algorithm. In each iteration, a new guess of discrete saturation $S^{k+1}_{n,i}$ is given (in the current time step $k + 1$) and the upstream saturation in (14) are recomputed.

4 Numerical experiments

In this section, we use the numerical scheme (13) to simulate the laboratory experiment that was carried out in the Center for Experimental Study of Subsurface Environmental

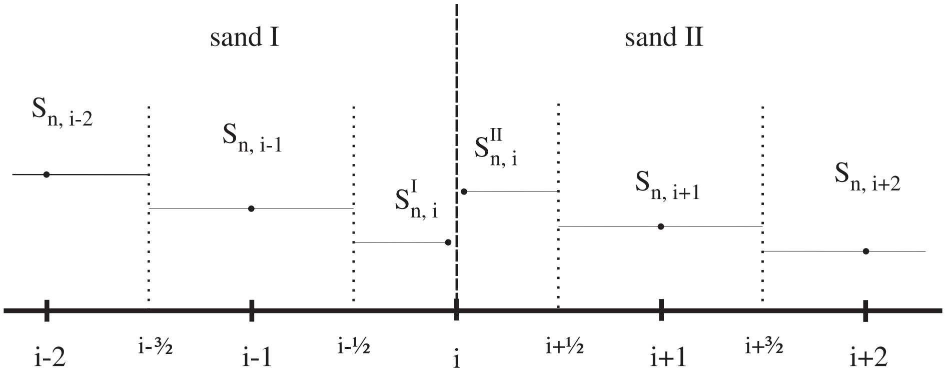


Figure 1: Discretization of the saturation jump at material discontinuity.

Processes, Colorado School of Mines. As a result of this experiment, three functional models of the dynamic effect coefficient $\tau = \tau(S_w)$ were correlated.

Models of the dynamic effect coefficient $\tau = \tau(S_w)$ were estimated as a result of the laboratory experiment, which consisted of a single, vertically placed, 10 cm long tube uniformly filled with a homogeneous sand. Initially, the column is flushed with water such that no air phase is present inside. A series of slow drainage steps was carried out in order to determine the capillary pressure – saturation relationship in equilibrium p_c^{eq}. The measured Brooks and Corey model parameters are shown in Table 2. Then, a series of fast drainage and imbibition experiments was performed and values of the capillary pressure and the air saturation are measured by probes in the middle of the column. Based on these measurements, three models of the dynamic effect coefficient τ were correlated, (see Table 3).

We simulate the experiment as a one-dimensional problem with different models of $\tau(S_w)$. The parameters of the discrete problem (13) are summarized in Table 1. The resulting temporal profiles of the air saturation S_n and the capillary pressure p_c are shown in Figure 2. In these numerical simulations, the measured outflow of water is used as a Neumann boundary condition at the bottom of the column ($x = L$).

The non-smooth shapes of the numerical solutions in Figure 2 are caused solely by the non-smoothness of the prescribed flux of water. Since the temporal derivative of the air saturation is directly influenced by the given flux, the non-smoothness is magnified in the values of the dynamic capillary pressure given by (11). That is why the bumps do not appear in the case of the static capillary pressure .

The influence of different models of the dynamic effect coefficient τ on the numerical solution of the air saturation S_n is negligible (see Figure 2). On the other hand, their influence on the capillary pressure p_c is important in cases, where there is a temporal change in the saturation S_n because the temporal derivative of S_n is multiplied by the dynamic effect coefficient τ in (11). The constant model for τ does not seem to be a good model for the appropriate approximation because its numerical solution of p_c differs substantially from the measured capillary pressure (see Figure 2). Therefore, the constant model requires further investigation of its validity.

Initial condition	$S_n(x,0) = 0$	$\forall x \in (0, L)$
Boundary conditions	$u_n(0,t) = 0$	$\forall t \in [0, T]$
	$p_n(0,t) = \text{const} = 0$	$\forall t \in [0, T]$
	$u_w(L,t) = \text{measured outflow}$	$\forall t \in [0, T]$
	$u_n(L,t) = 0$	$\forall t \in [0, T]$
Problem setup	$T = 5000\ s$, $L = 10\ cm$, $g = 9.81\ ms^{-2}$	
Capillary pressure	Dynamic capillary pressure p_c, various models for $\tau(S_w)$	
Sand	Ohji sand, Table 2	
Fluids	Air and water, Table 4	

Table 1: Parameters of the simulation of the laboratory experiment

5 Conclusions

A one-dimensional numerical scheme of two-phase incompressible and immiscible flow is presented that enables simulation of two-phase flow in homogeneous porous media under dynamic capillary pressure conditions.

Laboratory measured parameters were used in the numerical simulation of the dynamic capillary pressure including three models of the dynamic effect coefficient $\tau = \tau(S_w)$. The numerical solutions for the non-static capillary pressure show that the dynamic effect has a significant impact on the magnitude of the capillary pressure while the change in the saturation profiles may be considered negligible in some cases. The constant model of τ showed rather unrealistic profile of the numerical approximation of the capillary pressure when compared to the laboratory measured data.

Results of the simulation indicate that the dynamic effect may not be so important in drainage problems in a homogeneous porous medium. However, it may be of a great importance in highly heterogeneous media where the capillarity governs flow through material interfaces.

Parameter			Ohji sand
Porosity	Φ	$[-]$	0.448
Intrinsic permeability	K	$[m^2]$	$1.63{\cdot}10^{-11}$
Residual water saturation	S_{wr}	$[-]$	0.265
Brooks-Corey entry pressure	p_d	$[Pa]$	3450
Brooks-Corey pore size dist. index	λ	$[-]$	4.66

Table 2: Properties of porous media used in the numerical simulation.

Acknowledgement

This work has been supported by the project "Mathematical Modelling of Multiphase Porous Media Flow" 201/08/P567 of the Czech Science Foundation (GA ČR).

Model of τ $[Pa\ s]$	Ohji sand
Stauffer model	$\tau(S_w) = \tau_{S,Ohji} = 3.3 \cdot 10^5$
Constant model	$\tau_{Ohji}(S_w) = 1.1 \cdot 10^6$
Linear model	$\tau_{Ohji}(S_w) = 3.2 \cdot 10^6 (1 - S_w)$
Loglinear model	$\tau_{Ohji}(S_w) = 10^8 \exp(-7.7 S_w)$

Table 3: Experimentally determined models of the dynamic effect coefficient τ for the Ohji sand.

Parameter			Water	Air
Density	ρ	$[kg\ m^{-3}]$	997.8	1.205
Dyn. viscosity	μ	$[kg\ m^{-1} s^{-1}]$	$9.77 \cdot 10^{-4}$	$1.82 \cdot 10^{-5}$

Table 4: Fluid properties used in the simulations.

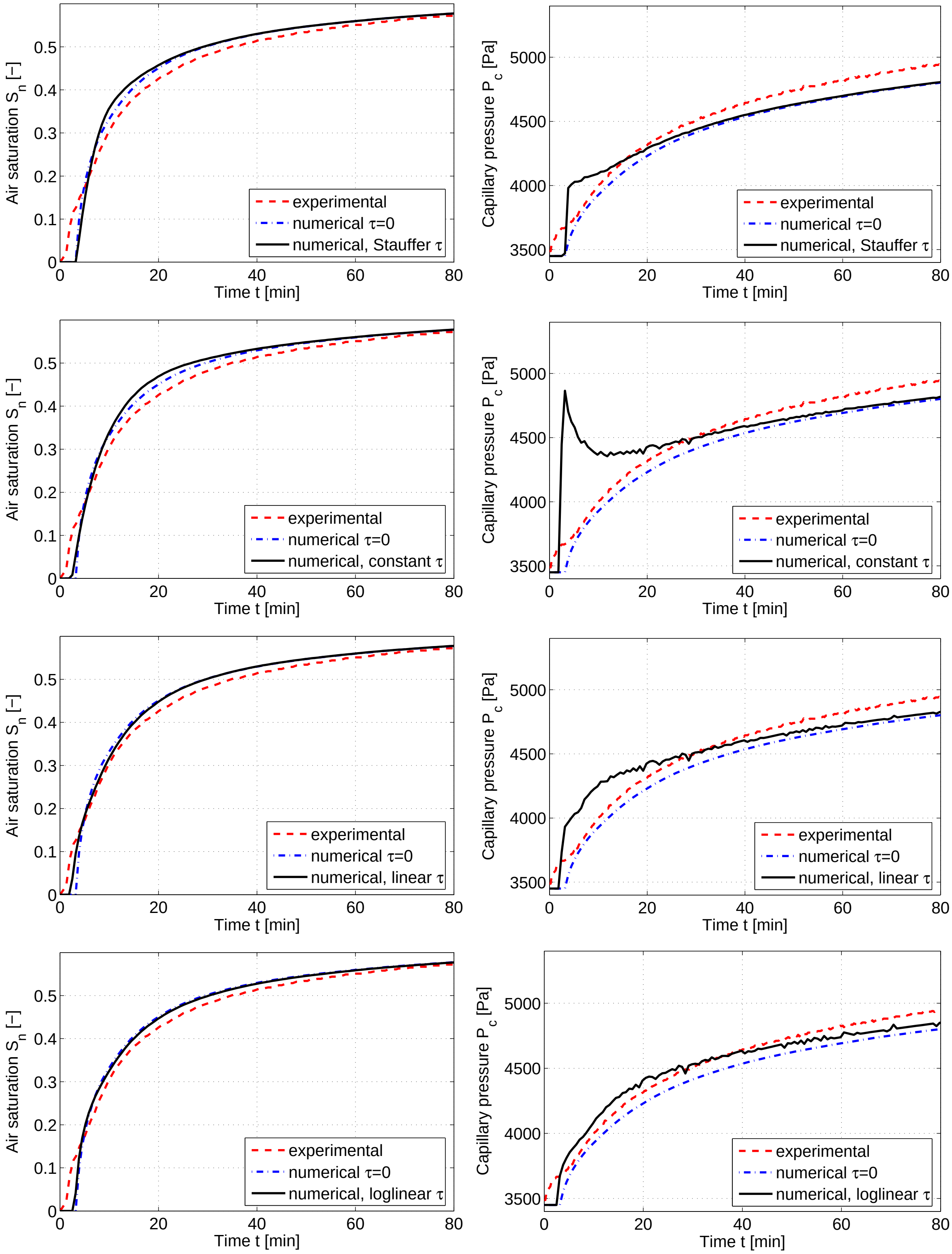


Figure 2: Numerical solutions and laboratory measured S_n and p_c in the middle of the column for various models of $\tau = \tau(S_w)$

References

[1] P. Bastian. *Numerical Computation of Multiphase Flows in Porous Media.* Habilitation Dissertation, Kiel university (1999).

[2] J. Bear and A. Verruijt. *Modeling Groundwater Flow and Pollution.* D. Reidel, Holland, Dordrecht, (1990).

[3] A. Beliaev and S. Hassanizadeh. *A Theoretical Model of Hysteresis and Dynamic Effects in the Capillary Relation for Two-phase Flow in Porous Media.* Transport in Porous Media **43** (2001), 487–510.

[4] R. H. Brooks and A. T. Corey. *Hydraulic properties of porous media.* Hydrology Paper 3 **27** (1964).

[5] N. Burdine. *Relative permeability calculations from pore-size distribution data.* Trans. AIME **198** (1953), 71–78.

[6] H. Dahle, M. Celia, and M. S. Hassanizadeh. *Bundle-of-Tubes Model for Calculating Dynamic Effects in the Capillary-Pressure-Saturation Relationship.* Transport in Porous Media **58** (2005), 5–22.

[7] D. Das, S. Hassanizadeh, B. Rotter, and B. Ataie-Ashtiani. *A Numerical Study of Micro-Heterogeneity Effects on Upscaled Properties of Two-Phase Flow in Porous Media.* Transport in Porous Media **56** (2004), 329–350.

[8] R. Fučík, M. Beneš, J. Mikyška, and T. H. Illangasekare. Generalization of the benchmark solution for the two-phase porous-media flow. In 'Finite Elements Models, MODFLOW, and More : Solving Groundwater problems', 181–184, (2004).

[9] R. Fučík, J. Mikyška, M. Beneš, and T. Illangasekare. *An Improved Semi-Analytical Solution for Verification of Numerical Models of Two-Phase Flow in Porous Media.* Vadose Zone Journal **6** (2007), 93–104.

[10] R. Fučík, J. Mikyška, M. Beneš, and T. Illangasekare. *Semianalytical Solution for Two-Phase Flow in Porous Media with a Discontinuity.* Vadose Zone Journal **7** (2008), 1001–1007.

[11] R. Fučík, J. Mikyška, and T. H. Illangasekare. *Evaluation of saturation-dependent flux on two-phase flow using generalized semi-analytic solution.* Proceedings on the Czech Japanese Seminar in Applied Mathematics (2004), 25–37.

[12] W. Gray and S. Hassanizadeh. *Paradoxes and Realities in Unsaturated Flow Theory.* Water Resources Research **27** (1991), 1847–1854.

[13] W. Gray and S. Hassanizadeh. *Unsaturated Flow Theory Including Interfacial Phenomena.* Water Resources Research **27** (1991), 1855–1863.

[14] S. Hassanizadeh, M. Celia, and H. Dahle. *Dynamic Effect in the Capillary Pressure-Saturation Relationship and its Impacts on Unsaturated Flow.* Vadose Zone Journal **1** (2002), 38–57.

[15] S. Hassanizadeh and W. Gray. *Thermodynamic basis of capillary pressure in porous media.* Water Resources Research **29** (1993), 3389–3406.

[16] R. Helmig. *Multiphase Flow and Transport Processes in the Subsurface : A Contribution to the Modeling of Hydrosystems.* Springer Verlag, Berlin, (1997).

[17] R. Helmig, A. Weiss, and B. Wohlmuth. *Dynamic capillary effects in heterogeneous porous media.* Computational Geosciences **11** (2007), 261–274.

[18] O. Ippisch, H. Vogel, and P. Bastian. *Validity limits for the van Genuchten-Mualem model and implications for parameter estimation and numerical simulation.* Advances in Water Resources **29** (2006), 1780–1789.

[19] S. Manthey. *Two-phase flow processes with dynamic effects in porous media - parameter estimation and simulation.* Institut fur Wasserbau der Universitat Stuttgart, Stutgart, (2006).

[20] S. Manthey, M. S. Hassanizadeh, and R. Helmig. *Macro-Scale Dynamic Effects in Homogeneous and Heterogeneous Porous Media.* Transport in Porous Media **58** (2005), 121–145.

[21] J. Mikyška, M. Beneš, and T. Illangasekare. *Numerical investigation of NAPL behavior at heterogeneous sand layers using VODA multiphase flow code.* to appear in Journal of Porous Media (2008).

[22] M. Peszyńska and S. Yi. *Numerical methods for unsaturated flow with dynamic capillary pressure in heterogeneous porous media.* International Journal of Numerical Analysis and Modeling **5** (2008), 126–149.

[23] T. Sakaki, D. O'Carroll, and T. Illangasekare. *Direct laboratory quantification of dynamic coefficient of a field soil for drainage and wetting cycles.* American Geophysical Union, Fall Meeting 2007, abstract# H53F-1486 (2007).

[24] F. Stauffer. Time dependence of the relations between capillary pressure, water content and conductivity during drainage of porous media. In 'On scale effects in porous media, IAHR, Thessaloniki, Greece', (1978).

[25] M. T. van Genuchten. *A closed-form equation for predicting the hydraulic conductivity of unsaturated soils.* Soil Science Society of America Journal **44** (1980), 892–898.

Vizualizace BTF textur v Blenderu

Martin Hatka

4. ročník PGS, email: hatka@utia.cas.cz
Katedra matematiky, Fakulta jaderná a fyzikálně inženýrská, ČVUT v Praze
školitel: Michal Haindl, ÚTIA, AV ČR

Abstract. Bidirectional texture function, also known as BTF texture, is 7D function of planar coordinates, spectral coordinate, and viewing and ilumination angles. In comparison with smooth textures, visual appearance of the BTF texture depends on viewing and illumination conditions. BTF textures are acquired by complex measurements of the real materials and subsequent image processing. Techniques from measurement to BTF texture rendering have been described well. On the other hand, there is no environment involving support to the BTF texture rendering. This paper describes novel Blender texture plugin for purpose of BTF texture mapping and rendering. Our previously developed BTF Roller algorithm is also implemented in the plugin. Described plugin allows to create realistic computer animations with additional BTF textures of desired size mapped onto an object surfaces while the other functionality of Blender retains.

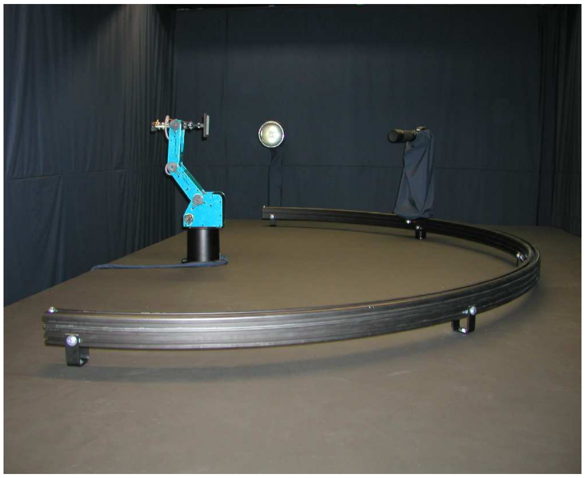
Abstrakt. Bidirectional texture function, známá též jako BTF textura, je sedmirozměrná funkce planárních souřadnic, spektrální souřadnice, úhlů pohledu a osvětlení. Oproti hladkým texturám zachycuje závislost vzhledu reálných materiálů na světelných a pozorovacích podmínkách. BTF textury jsou získávány složitým měřením reálných materiálů a následným zpracováním naměřených dat. Techniky od měření po renderování BTF textur jsou již popsány v odborné literatuře. Na druhou stranu však chybí dostupné prostředí, ve kterém by byla implementována podpora pro tento typ textur. Tento článek prezentuje rozšíření 3D grafického editoru Blender o podporu BTF textur pomocí texturního modulu, včetně implementovaného algoritmu BTF Roller pro syntézu BTF textur. Popsaná implementace umožňuje tvorbu realistických počítačových animací s využitím BTF textur libovolných rozměrů mapovaných na povrchy objektů a současně zachovává plnou funkcionalitu Blenderu.

1 Úvod

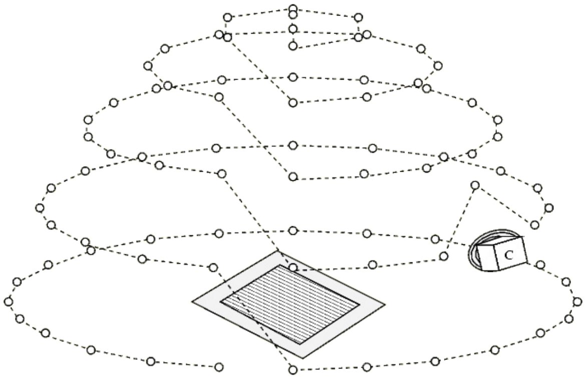
1.1 BTF

Textura reprezentuje vizuální vlastnosti materiálu jako funkce planárních souřadnic (x, y), případně 3D textura jako funkce prostorových souřadnic (x, y, z). BTF textura [2] je závislá na světelných a pozorovacích podmínkách, jde o 7D funkci planárních a spektrálních souřadnic a úhlů osvětlení a pohledu, $\text{BTF}(x, y, r, \theta_l, \phi_l, \theta_v, \phi_v)$. Existuje několik systémů pro měření BTF textur [2, 8] a od nich se odvíjí způsob jejich reprezentace a množství uložených dat. Jednou z možností je reprezentace BTF textury jako množiny několika tisíců vzájemně registrovaných planárních textur, další možností je uložení takové textury ve formátu HDR.

V tomto článku jsou využívány BTF textury z měřícího zařízení na Univerzitě v Bonnu (obr. 1), které v současnosti představuje nejlepší měřící zařízení na světě. Taková BTF textura se pak skládá z 6561 barevných obrázků, což je kombinací 81 pozic kamery a 81 pozic světelného zdroje (obr. 2).



Obrázek 1: Robot pro měření BTF textur na Univerzitě v Bonnu. Vzorek materiálu se naklápí, světelný zdroj i kamera se pohybuje, změří se 6561 kombinací.



Obrázek 2: Znázorněno je 81 pozic pro světelný zdroj i pro kameru. Během měření BTF textury je měřeno 6561 (81×81) kombinací polohy světelného zdroje a kamery.

Existují také způsoby pro reprezentaci BTF prostoru pomocí matematických modelů [3, 7, 4, 6], ovšem ty jsou nad rámec tohoto článku.

1.2 Vizualizace BTF

Ve světě zatím neexistuje žádný profesionální nebo veřejný software pro vizualizaci BTF textur. Na druhou stranu naprogramování celého nezbytného grafického prostředí by bylo velmi náročné a s největší pravděpodobností by nebylo dosaženo takových možností, jaké nabízí současné nástroje pro tvorbu 3D grafiky a animací. Proto bylo potřeba najít aplikaci, která by šla vhodně rozšířit o vizualizaci BTF. Jako ideální volbou pro implementaci vizualizace BTF se ukázal Blender, který je v současné době aktivně vyvíjen sdružením Blender Foundation a je volně dostupný včetně zdrojových kódů.

1.3 Blender

Blender je open-source software pro modelování a vykreslování 3D počítačové grafiky a animací s využitím různých technik, jako např. raytracing, radiosita, scanline rendering. Vlastní uživatelské rozhraní je vykreslováno pomocí knihovny OpenGL. OpenGL umožňuje nejen hardwarovou akceleraci vykreslování 2D a 3D objektů, ale především snadnou přenositelnost na všechny podporované platformy, např. FreeBSD, IRIX, GNU/Linux, Microsoft Windows, Mac OS X a Solaris.

Modelovací schopnosti jsou zaměřeny především na práci s ploškovou reprezentací těles. Blender umožňuje pracovat s takzvanými subsurf plochami, dále pak podporuje v omezenější formě práci s parametrickými plochami a křivkami (Bezier, NURBS) a implicitními plochami (MetaBalls).

Animační možnosti nejsou omezeny pouze na jednoduché klíčování objektů a jejich tvarů, ale Blender umožňuje animovat objekty i pomocí armatur a inverzní kinematiky.

Dále má implementovánu podporu pro fluidní dynamiku, softbodies, různé deformátory, částicové sytémy, apod.

1.4 Texturní moduly v Blenderu

Blender má v sobě zabudovánu podporu pro několik typů textur, z nichž důležitá je podpora rastrových obrázků, které lze následně mapovat na libovolné objekty jako textury. Zde je možné využít standardních formátů BMP, PNG, JPEG a mnoho dalších. Nejdůležitější možností pro renderování BTF jsou texturní moduly (pluginy). Texturním modulem je zde dynamicky linkovaná knihovna, v níž jsou obsaženy funkce s pevně definovanými hlavičkami, zejména pak funkce, která vrací barvu požadovaného pixelu s danými texturními souřadnicemi. Když je třeba znát barvu takového pixelu, zavolá Blender příslušnou funkci z modulu, jíž jsou texturní souřadnice předávány jako parametry. Tato funkce pak vrátí barvu pixelu.

2 BTF texturní modul

Vzhledem k tomu, že BTF textura je množina několika tisíc barevných obrázků, bylo by velmi komplikované implementovat podporu pro BTF textury přímo do Blenderu. Právě možnosti texturního modulu se ukázaly jako nejvýhodnější. Navržený BTF modul komunikuje s Blenderem pomocí standardního rozhraní, které bylo třeba rozšířit pouze o předávání úhlů pohledu a osvětlení modulu. Veškeré další výpočty a operace s BTF texturami jsou čistě v režii modulu.

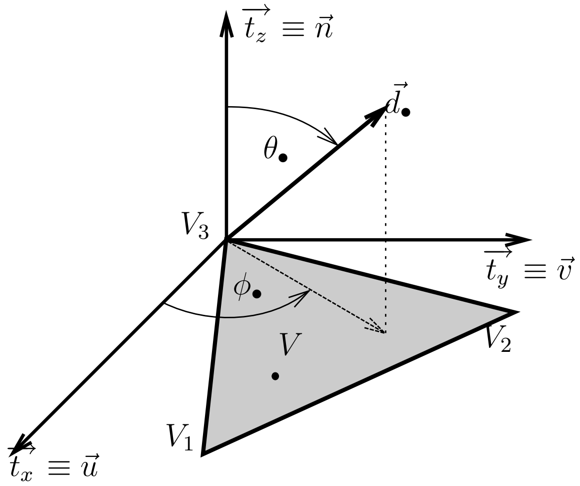
Tento způsob využití BTF v Blenderu má několik dalších zásadních výhod. Hlavní z nich je možnost implementace různých algoritmů pro syntézu textur a BTF textur přímo v 3D grafickém editoru, zejména pak dříve vyvinutého algoritmu BTF Roller [5]. Další výhodou je mnohem snažší následná optimalizace výkonu modulu bez nutnosti zásahu do rendereru Blenderu.

2.1 Rozhraní pro texturní moduly

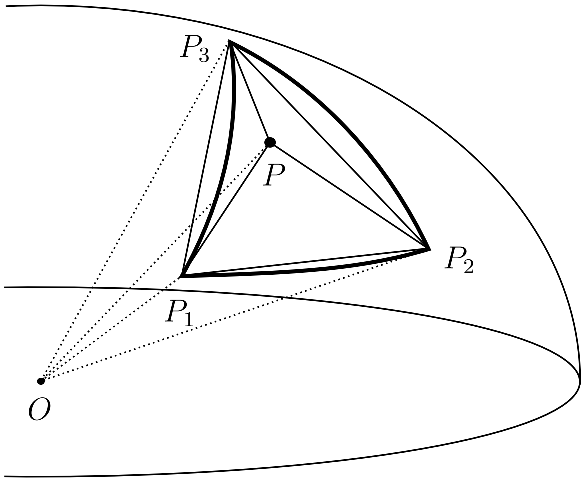
Vstupem pro texturní modul jsou prostorové texturní souřadnice (u, v, w), výstupem je pak vektor $(i, Y_R, Y_G, Y_B, N_u, N_v, N_w)$, kde i je intenzita (v případě monochromatické textury), Y_R, Y_G a Y_B jsou složky RGB a N_u, N_v a N_w jsou složky vektoru normály v texturních souřadnicích.

Využití všech složek vstupního i výstupního vektoru je nepovinné, pro texturní modul pro BTF textury byla využita pouze dvojice texturních souřadnic (u, v) na vstupu a trojice (Y_R, Y_G, Y_B) na výstupu.

Automatickým generování texturních souřadnic s využitím pomocných těles (krychle, válec, koule) nelze pro složité povrchy objektů dosáhnout kvalitních výsledků. Blender však nabízí propracované nástroje pro ruční mapování textur technikou nazývanou UV-mapování nebo UV-mapping. UV-mapování přiřazuje libovolnému bodu povrchu objektu texturní souřadnice, jde tedy o mapovací funkci T_{UV}, $(u, v) = T_{UV}(x, y, z)$. Současně se jedná o nejpřesnější způsob mapování.


$\vec{t_z} \equiv \vec{n}$
$\vec{d}$
θ
V_3
$\vec{t_y} \equiv \vec{v}$
ϕ
V
V_2
V_1
$\vec{t_x} \equiv \vec{u}$

Obrázek 3: Výpočet azimutu a elevace pro vektory pohledu a osvětlení. Je založen na transformaci vektoru pohledu a osvětlení do prostoru texturních souřadnic.



Obrázek 4: Interpolace motivovaná sférickými barycentrickými souřadnicemi. Příspěvky od P_1, P_2, P_3 k P odpovídají objemům protějších čtyřstěnů.

Jak bylo uvedeno výše, BTF textura je 7D funkcí planárních souřadnic, spektrální souřadnice a úhlu pohledu a osvětlení, $\mathrm{BTF}(x, y, r, \theta_l, \phi_l, \theta_v, \phi_v)$. Vstupní rozhraní modulu tedy muselo být rozšířeno o předávání úhlu pohledu a úhlu osvětlení, vstupní vektor (u, v, w) byl nahrazen vstupním vektorem $(u, v, w, \theta_l, \phi_l, \theta_v, \phi_v)$, kde složka w není využita.

Součástí rozhraní je možnost definovat pomocí standardních ovládacích prvků uživatelského rozhraní Blenderu ovládací panel modulu, který je k dispozici při jeho použití a umožňuje modulu předávat další parametry pro jeho ovládání. Konkrétní parametry předávané BTF modulu budou popsány později.

2.2 Podpora Blenderu pro BTF textury

Blender neuvažuje ve svém rendereru závislost textury na úhlu pohledu a úhlu osvětlení. Implementováno je sice několik typů difúzních a spekulárních shaderů, které pracují s normálou povrchu, vektorem pohledu a vektorem osvětlení, nicméně texturování probíhá dříve a mimo shadery a jejich využití ztrácí pro potřeby BTF význam.

Pro mapování BTF textur na povrchy objektů a jejich renderování nejsou shadery ve skutečnosti potřeba, protože efekty vznikající osvětlením jsou zachyceny v naměřených BTF datech.

Jedinou nutnou úpravou bylo rozšíření rendereru o podporu výpočtu úhlů elevací θ_v resp. θ_l a azimutů ϕ_v resp. ϕ_l pro pohled resp. osvětlení (obr. 3).

Renderer Blenderu vykresluje danou scénu bod po bodu a při vykreslování konkrétního bodu, kterému odpovídá bod nějakého objektu, pracuje s vektorem pohledu a s vektory příchozího osvětlení. Vektorů pro osvětlení je tolik, kolik je světelných zdrojů. Výsledné osvětlení bodu scény je dáno součtem příspěvků od jednotlivých světelných zdrojů. Pro každý vektor osvětlení je tedy nutné určit úhly θ_l, ϕ_l, θ_v a ϕ_v, na nichž závisí příspěvky barvy renderovaného pixelu od jednotlivých světelných zdrojů.

2.2.1 Výpočet úhlů pohledu a osvětlení v rendereru

Každý renderovaný pixel náleží renderovanému trojúhelníku. Uvažujme vykreslovaný bod scény, kterému odpovídá bod V na povrchu určitého objektu (obr. 3). Tento objekt je reprezentován množinou trojúhelníkových plošek, z nich označme $\triangle V_1V_2V_3$ trojúhelníkovou plošku, která obsahuje bod V. Označme $(V_1)_{\mathcal{S}} = (v_1^x, v_1^y, v_1^z)$, $(V_2)_{\mathcal{S}} = (v_2^x, v_2^y, v_2^z)$, $(V_3)_{\mathcal{S}} = (v_3^x, v_3^y, v_3^z)$, kde $\mathcal{S} = (\vec{x}, \vec{y}, \vec{z})$ je ortonormální bází prostorových souřadnic, souřadnice bodů V_1, V_2, V_3. Mapovací funkce T_{UV}, $T_{UV}(v_\bullet^x, v_\bullet^y, v_\bullet^z) = (v_\bullet^u, v_\bullet^v)$, jim pak přiřazuje texturní souřadnice $(V_1)_{\mathcal{T}} = (v_1^u, v_1^v, 0)$, $(V_2)_{\mathcal{T}} = (v_2^u, v_2^v, 0)$, $(V_3)_{\mathcal{T}} = (v_3^u, v_3^v, 0)$, kde $\mathcal{T} = (\vec{u}, \vec{v}, \vec{w})$ je ortonormální bází třírozměrného prostoru texturních souřadnic. Poznamenejme, že ploška $\triangle V_1V_2V_3$ leží v rovině dané osami $\vec{u}$ a $\vec{v}$ a její normála $\vec{n}$ je rovnoběžná s osou $\vec{w}$. Ze znalosti vztahů mezi prostorovými a texturními souřadnicemi se vyjádří osy texturních souřadnic v souřadnicích prostorových, tj. $\vec{t_x} = (\vec{u})_{\mathcal{S}}$, $\vec{t_y} = (\vec{v})_{\mathcal{S}}$ a $\vec{t_z} = (\vec{w})_{\mathcal{S}}$.

Označme vektory $\vec{d_v}$ resp. $\vec{d_l}$ pro směr pohledu resp. osvětlení vyjádřené v prostorových souřadnicích. Projekcí vektoru $\vec{d_\bullet}$ na osy $\vec{t_x}$, $\vec{t_y}$, $\vec{t_z}$ texturních souřadnic vyjádřených v prostorových souřadnicích pak získáme vyjádření vektoru $\vec{d_\bullet}$ v texturních souřadnicích $\vec{s_\bullet} = (s_\bullet^u, s_\bullet^v, s_\bullet^w) = (\vec{d_\bullet})_{\mathcal{T}}$. Konečně z vyjádření vektoru $\vec{s_\bullet} = (\vec{d_\bullet})_{\mathcal{T}}$ v texturních souřadnicích vypočteme potřebné úhly $\theta_\bullet$, $\phi_\bullet$ pomocí vztahů mezi kartézskými a sférickými souřadnicemi

$$\cos\theta = s^w, \qquad \sin\phi = \frac{s^v}{\sqrt{(s^u)^2 + (s^v)^2}}, \qquad \cos\phi = \frac{s^u}{\sqrt{(s^u)^2 + (s^v)^2}}.$$

2.3 Implementace BTF modulu

Texturní modul je dynamicky linkovaná knihovna obsahující funkci s hlavičkou definovanou dle rozhraní pro texturní moduly, která je opakovaně volána během renderování scény. Jako parametry jsou ji předávány texturní souřadnice a úhly pohledu a osvětlení, funkce vrací barvu pixelu pro zadané vstupní parametry.

BTF textura je měřena pro 81 různých pozic kamery v kombinaci s 81 pozicemi světelného zdroje. V praxi je ale potřeba vypočítat barvu pixelu pro jiné než naměřené kombinace úhlů. Pro néznámé úhly je barva pixelu iterpolována z nejbližších známých pozic principem, který je motivován sférickými barycentrickými souřadnicemi [9].

Protože se jedna BTF textura skládá z 6561 barevných obrázků, nebylo by efektivní celou texturu načítat do paměti. Jako dostačující se ukázalo implementovat zásobník načtených obrázků pro potřebné kombinace úhlů pohledu a osvětlení, přičemž uživatel modulu má možnost ovlivnit velikost zásobníku.

Posledním důležitým krokem při návrhu modulu byla implementace algoritmu BTF Roller [5] pro syntézu BTF textur. Vstupní data pro modul musí být ve formě vzájemně zaměnitelných texturních dlaždic, které jsou nalezeny během analytické části algoritmu.

2.3.1 Interpolace barycentrickými vahami

BTF textura je měřena pouze pro dané kombinace úhlů pohledu a osvětlení, navíc 81 pozic světelného zdroje i kamery, vytvářející polokouli nad měřeným vzorkem, současně definuje triangulaci polokoule.

V praxi je při renderování BTF povrchů objektů nutné znát barvu pixelu pro libovolnou kombinaci pohledu a osvětlení. Vzhledem ke známé triangulaci se přirozeně nabízí interpolace, která vychází z principu sférických barycentrických souřadnic [9]. Interpolace s využitím sférických barycentrických souřadnic by byla nejpřesnější, na druhou stranu ale výpočetně náročná. Následující aproximace se však ukázala jako dostatečná.

Předpokládejme nyní polokouli (obr. 4) se středem O, na ní bod P odpovídající požadovanému azimutu a elevaci pro pohled nebo osvětlení, dále označme P_1, P_2 a P_3 tři nejbližší známe body s naměřenými hodnotami Y_{P_1}, Y_{P_2} a Y_{P_3} a w_1, w_2 a w_3 jejich příspěvky pro Y_P, tj. $Y_P = w_1 Y_{P_1} + w_2 Y_{P_2} + w_3 Y_{P_3}$, $w_1 + w_2 + w_3 = 1$. S využitím sférických barycentrických souřadnic by byly váhy w_1 resp. w_2 resp. w_3 rovny obsahu sférického trojúhelníku $\triangle PP_2P_3$ resp. $\triangle PP_3P_1$ resp. $\triangle PP_1P_2$ dělené obsahem sférického trojúhelníku $\triangle P_1P_2P_3$. Upuštěním od sférických barycentrických souřadnic lze váhy w_1, w_2 a w_3 definovat následovně:

$$w_1 = \frac{V_1}{V}, \quad w_2 = \frac{V_2}{V}, \quad w_3 = \frac{V_3}{V}, \quad V = V_1 + V_2 + V_3,$$

kde V_1 je objem čtyřstěnu PP_2P_3O, V_2 objem PP_3P_1O, V_3 objem PP_1P_2O. Navíc když $O = (0,0,0)$, je $V_1 = \frac{1}{6}\,|\det(P, P_2, P_3)|$, $V_2 = \frac{1}{6}\,|\det(P, P_3, P_1)|$, $V_3 = \frac{1}{6}\,|\det(P, P_1, P_2)|$.

2.3.2 Zásobník pro BTF data

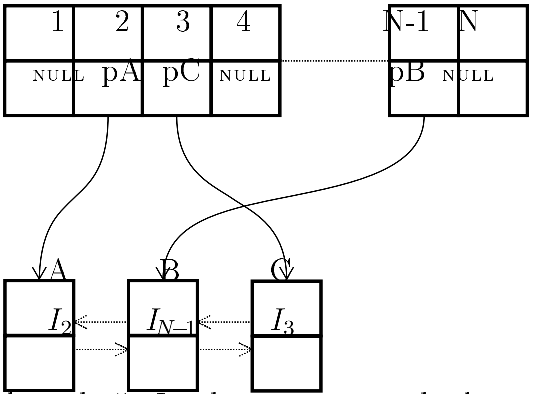
Texturní modul je navržen a implementován tak, že každá jeho instance používá vlastní sadu parametrů a obrazová data. Aby nebylo během renderování zbytečně plýtváno operační pamětí počítače, jsou načtená obrazová data, kde každá kombinace úhlu pohledu a osvětlení má unikátní index, uložena v zásobníku. Zásobník je implementován jako dvojsměrný seznamu o velikosti definované uživatelem (obr. 5). Navíc počet obrazů v BTF textuře je 6561, proto je součástí zásobníku pole ukazatelů na prvky seznamu indexované indexy jednotlivých obrazů. Díky tomu lze velmi efektivně, bez procházení seznamu, zjistit, zda je požadovaný obraz v seznamu.

V okamžiku, kdy texturní modul potřebuje obraz s daným indexem, zjistí, zda je aktuálně v paměti. Pokud ne, načte obrazová data. Aby bylo načítání resp. odstraňování obrazů do resp. z paměti efektivní, je aktuálně použitý obraz s daným indexem přesunut na začátek seznamu. Nejdéle nepoužitý obraz je vždy na konci seznamu. Je-li seznam naplněn, je před vložením dalšího obrazu nejprve odstraněn obraz z jeho konce.

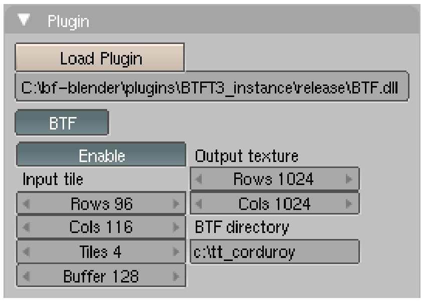
2.3.3 Syntéza BTF textur

Aby syntetická výstupní textura modulu mohla mít libovolné požadované rozměry a aby vypadala stále realisticky jako původní textura, byl do modulu implementován algoritmus BTF Roller [5] pro syntézu textur v reálném čase. Přesněji, byla implementována pouze část syntézy, vstupní data pro modul jsou výstupem analytické části algoritmu. Pro rozhodnutí, která texturní dlaždice se má pro renderovaný pixel použít, byl využit princip Wangových dlaždic [1].

Použitím Wangových dlaždic lze získat deterministický předpis pro aperiodické vydláždění libovolně velké plochy konečnou množinou malých dlaždic. Pro potřeby texturního modulu byla využita iterativní varianta publikovaná v [10]. Při inicializaci modulu



Obrázek 5: Implementace zásobníku pro načítání obrázků s BTF daty. Dole dvojsměrný seznam obsahující obrazová data, nahoře pole ukazatelů na prvky seznamu.



Plugin

Load Plugin

C:\bf-blender\plugins\BTFT3_instance\release\BTF.dll

BTF

Enable

Input tile

Rows 96

Cols 116

Tiles 4

Buffer 128

Output texture

Rows 1024

Cols 1024

BTF directory

c:\tt_corduroy

Obrázek 6: Rozhraní BTF texturního modulu. Pro správnou funkci je třeba nastavit velikost a počet dlaždic, velikost vyrovnávací paměti, požadovanou velikost syntetické textury a adresář s umístěním textury na disku.

je vypočten počet potřebných iterací (v řádu jednotek), při syntéze je pak pro každý renderovaný pixel určena dlaždice, která má být použita.

2.3.4 Parametry modulu

Pro požadovaný vzhled BTF textury je třeba nastavit několik základních parametrů (obr. 6), zejména počet dlaždic a požadovanou velikost syntetické textury.

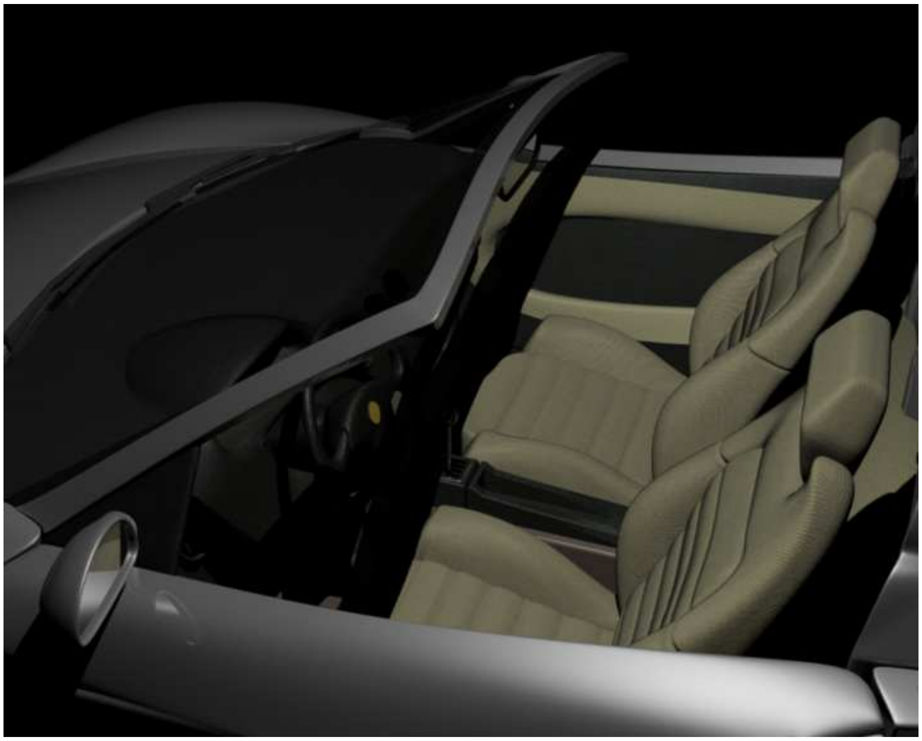
3 Výsledky

Texturní modul spolu s úpravou jádra rendereru Blenderu umožňuje použití BTF textur přímo v prostředí Blenderu. Použitím zde rozumíme jejich UV-mapování na 3D modely a následné fyzikálně správné, realistické zobrazení povrchů objektů s BTF materiály.

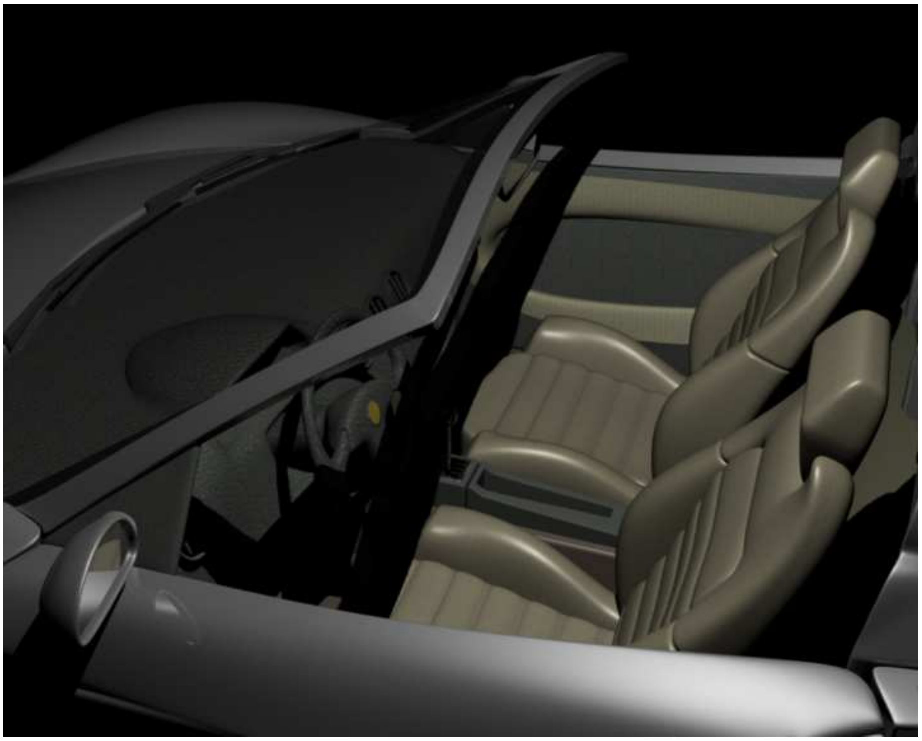
Obrázek 7 ilustruje použití vyvinutého texturního modulu při mapování a zobrazení BTF textur v interiéru modelu automobilu. Pro srovnání lze srovnat s obrázkem 8 se stejnou scénou, ve které byly místo BTF textur použity hladké textury. Na první pohled je patrné, že pomocí hladkých textur nelze, na rozdíl od BTF textur, modelovat efekty jako odlesky a změnu barvy materiálu v závislosti na pozorovacích a světelných podmínkách. Na obrázku 8 si lze všimnout nereálných odlesků na sedadlech, která jsou potažena látkou. Ještě lépe je rozdíl mezi mapováním a renderováním hladkých a BTF textur patrný na detailnějším obrázku 10, kde bylo pro srovnání použito více materiálů.

Pro vyšší rychlost při manipulaci s BTF texturami byl do texturního modulu implementován algoritmus BTF Roller [5], který v současnosti představuje nejrychlejší způsob syntézy BTF textur.

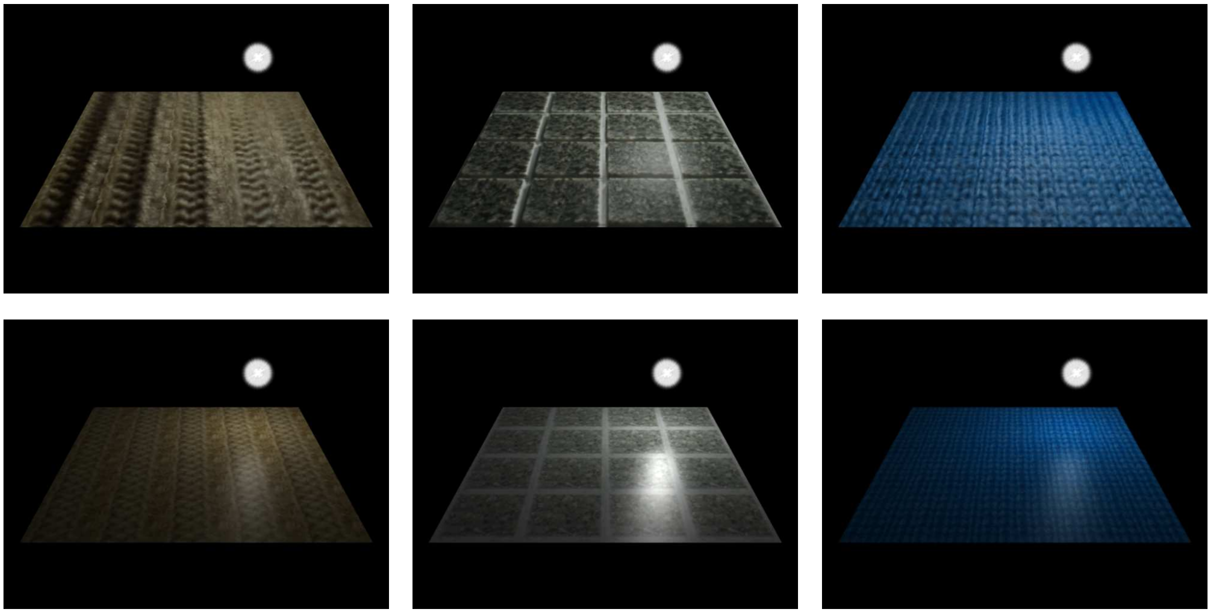
Použití BTF textur v prostředí Blenderu nijak výrazně nezpomaluje proces renderování. Nejvíce času spotřebuje texturní modul pro načítání jednotlivých kombinací pohledu a osvětlení BTF textury z pevného disku počítače. V první verzi, bez optimalizace zásobníkem popsaným v kapitole 2.3.2, byla manipulace s texturními daty časově velmi náročná. Po optimalizaci se čas na vykreslení scény zkrátil na 10% původního času.



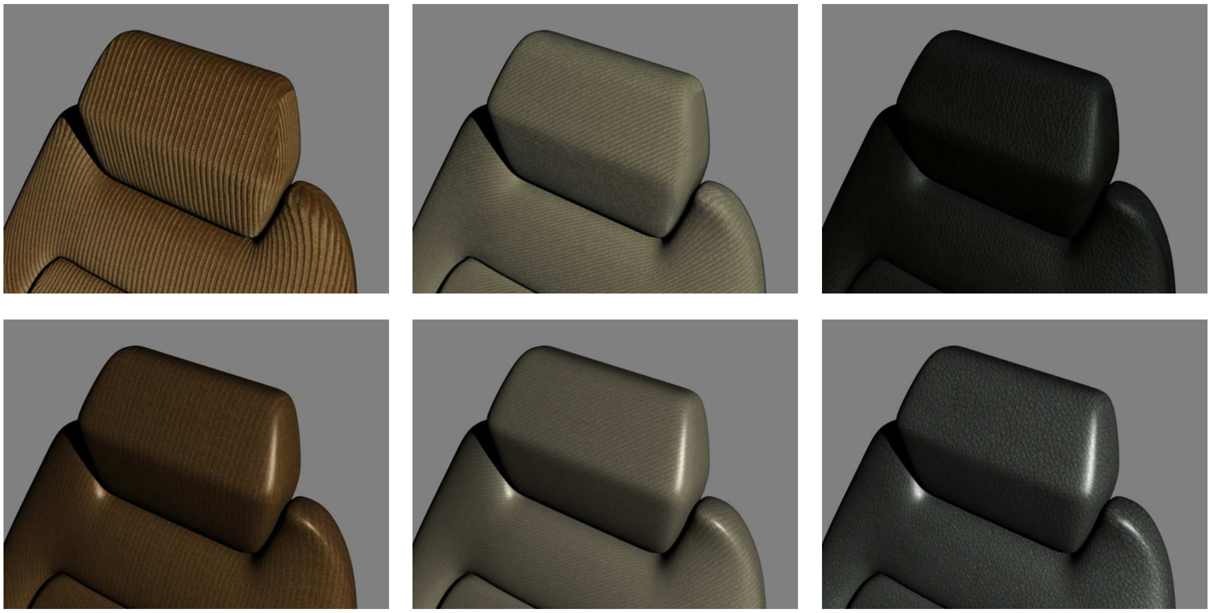
Obrázek 7: Model automobilu s BTF texturami použitými v interieru. Díky BTF texturám lze dosáhnout realistického vzhledu modelovaných objektů.



Obrázek 8: Model automobilu s hladkými texturami použitými v interieru. Mapováním hladkých textur nelze dosáhnout realistického vzhledu jako v případě BTF textur.



Obrázek 9: Porovnání aplikace BTF textur a hladkých textur. Zobrazeny jsou výsledky vizualizace testovací scény včetně umístění světelného zdroje. V horní řadě jsou mapovány BTF materiály, v dolní řadě pak odpovídající hladké textury. Je zřejmé, že v případě BTF textur vypadají materiály skutečně realisticky. Další komplikací je v případě hladkých textur správné nastavení shaderů pro jednotlivé materiály, v případě BTF použití shaderů odpadá.



Obrázek 10: Porovnání výsledků aplikace BTF textur a hladkých textur. V horní řadě zobrazena opěrka a část sedadla s namapovanými BTF texturami, v dolní řadě odpovídající planární textura. Zleva postupně použity materiály manšestr, látka a černá kůže.

4 Závěr

Texturní modul pro podporu BTF textur je významným přínosem pro oblast počítačové grafiky zabývající se zpracováním a vizualizací BTF textur. Díky němu se rozšiřují možnosti využití a prezentace výsledků různých metod modelující a zpracovávajících BTF textury.

Implementace texturního modulu je prvním krokem pro praktické využívání BTF textur v libovolných již existujících 3D modelech a 3D scénách. V další práci je plánováno rozšíření BTF texturního modulu i na UNIXové a linuxové platformy, dále rozšíření o podporu vícejádrových procesorů a především zobecnění modulu i pro další metody pro syntézu BTF textur a další BTF modely vyvinuté v ÚTIA.

Literatura

[1] M. F. Cohen, J. Shade, S. Hiller, and O. Deussen. *Wang tiles for image and texture generation.* ACM Trans. Graph. **22** (July 2003), 287–294.

[2] K. J. Dana, B. van Ginneken, S. K. Nayar, and J. J. Koenderink. *Reflectance and texture of real-world surfaces.* ACM Transactions on Graphics **18** (1999), 1–34.

[3] J. Filip and M. Haindl. *Bidirectional texture function modeling: A state of the art survey.* IEEE Transactions on Pattern Analysis and Machine Intelligence **31** (2009), 1921–1940.

[4] J. Filip and M. Haindl. *Non-linear reflectance model for bidirectional texture function synthesis.* Pattern Recognition, International Conference on **1** (2004), 80–83.

[5] M. Haindl and M. Hatka. BTF roller. In 'Texture 2005: Proceedings of 4th Internatinal Workshop on Texture Analysis and Synthesis', M. Chantler and O. Drbohlav, (eds.), 89–94, Edinburgh, (October 2005). Heriot-Watt University.

[6] M. Haindl and J. Filip. *Extreme compression and modeling of bidirectional texture function.* IEEE Transactions on Pattern Analysis and Machine Intelligence **29** (2007), 1859–1865.

[7] M. Haindl, J. Filip, and M. Arnold. *Btf image space utmost compression and modelling method.* Pattern Recognition, International Conference on **3** (2004), 194–197.

[8] D. Lyssi. A reflectometer setup for spectral btf measurement. In 'CESCG', (April 2009).

[9] K. Polthier, A. Belyaev, A. S. (editors, T. Langer, E. Belyaev, H. peter Seidel, and M. Informatik. Spherical barycentric coordinates, (2006).

[10] J. Stam. Aperiodic texture mapping. Technical report, European Research Consortium for Informatics and Mathematics (ERCIM), (1997).

Numerical Simulation of Spiral Growth by Phase-Field Method*

Hung Hoang Dieu

2nd year of PGS, email: hoangdieu@fjfi.cvut.cz
Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, CTU in Prague
advisor: Michal Beneš, Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, CTU in Prague

Abstract. In the paper we consider the problem of spiral crystal growth. This problem is described by a phase-field model based on the Burton-Cabrera-Frank theory (see Ref. [3]). For numerical simulations performed in three dimensions, we develop a numerical scheme based on the finite difference method. We investigate the influence of numerical parameters on the growth patterns. We present computational studies related to the pattern formation and to the dependence on model parameters.

Abstrakt. Příspěvek se zabývá problémem spirálového růstu reálných krystalů. Tento problém je popsán modelem typu phase-field založeným na Burtonově, Cabrerově a Frankově teorii (viz. [3]). Pro numerické simulace jsme vyvinuli numerické schéma, které je založené na metodě konečných diferencí. Zkoumáme vliv numerických parametrů na mechanismus růstu. Nakonec jsou prezentovány získané výsledky.

1 Introduction

There are two fundamental models of crystal growth mechanism: two-dimensional nucleation and layer growth of perfect crystals or spiral growth of real crystals (see Ref. [12], Chapter 3).

Real crystals contain dislocations which are crystallographic defects in the structure of the crystal lattice. The presence of dislocations influences the mechanism of crystal growth. If a screw dislocation is present in the crystal lattice of the substrate, a step with a zero height at the dislocation core is created. This step winds around the dislocation and produce a spiral (see Fig. 1).

Recently, crystal growth has been investigated from a mathematical point of view. For more details, we refer the reader to the works of Guo-Nakamura-Ogiwara-Tsai [5] and Ohtsuka [9].

*The work was partly supported by the project No. MSM 6840770010 "Applied Mathematics in Technical and Physical Sciences", and by the project No. LC06052 "Jindřich Nečas Center for Mathematical Modelling", both of the Ministry of Education, Youth and Sports of the Czech Republic.

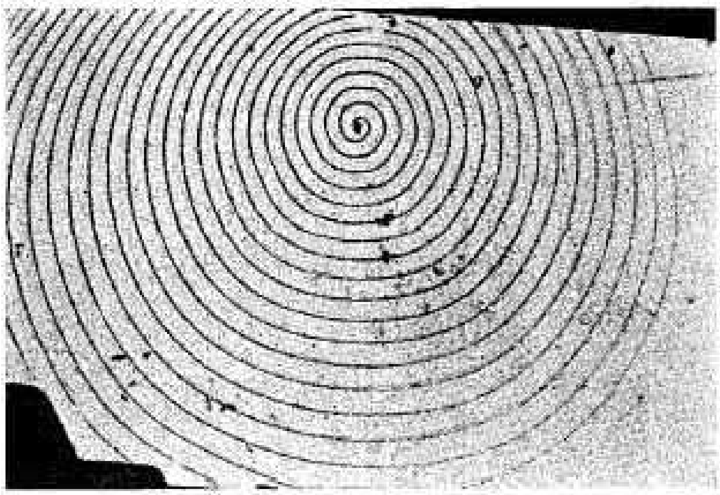


Figure 1: Growth spiral on a carborundum face. Crystallographic Laboratory, Geological Institute, University of Ghent.

2 The model

Classically epitaxial crystal growth is modeled using Burton-Cabrera-Frank (BCF) theory (see Ref. [1]). According to that theory atoms are first adsorbed to the crystalline surface. Such atoms are called adatoms. Then they diffuse freely along the surface and they can either desorb from the surface with a probability $1/\tau_S$ per unit time, or they are incorporated into the crystal at one of the three sites: ledge site, step site or kink site. Incorporation at a kink site will be the most energetically favorable. Two-dimensional growth occurs only at relatively higher super-saturation when random nuclei are generated on existing flat surface.

The BCF model consists of a diffusion equation for the concentration of adatoms, as well as two boundary conditions at the growing steps:

$$\partial_t c = D\Delta c - \frac{1}{\tau_S}c + F, \tag{1}$$

in the domain S and

$$c = c_{eq}(1 + \kappa\Omega\gamma/k_B T), \tag{2}$$

$$v_n = D\Omega[\frac{\partial c}{\partial n+} - \frac{\partial c}{\partial n-}], \tag{3}$$

on the interface $\Gamma(t)$. Here, c is the density of adatoms on the surface S, D is the surface diffusion coefficient, τ_S is the mean time for the desorption of adatoms from to the solution, F is the deposition rate, c_{eq} is the equilibrium concentration for a straight step, κ is the curvature of step $\Gamma(t)$, Ω is the area of a single atom, γ is the step stiffness, $k_B T$ describes the thermal energy for a fixed temperature T and v_n is the normal velocity of the step and $\frac{\partial u}{\partial n\pm}$ is the normal concentration gradient on the lower $(+)$ and upper $(-)$ side of the step.

Direct numerical simulations of the sharp-interface problem $(1) - (3)$ are difficult, since the position of the step has to be tracked explicitly (see Ref. [3]). The BCF model described above can be replaced by a phase field model where a higher-dimensional order parameter function $\Phi(x, y, t)$ is introduced whose values indicates the phase at a given position. In our case, the phase field $\Phi(x, y, t)$ describes the height of the epitaxial solid by the number of monoatomar layers. The phase-field model was previously used by Liu and Metiu [7] for one-dimensional step train, then enhanced by Karma and Plapp [5]. This model, which represents a system of parabolic partial differential equations, has the form

$$\begin{aligned} \partial_t c &= D\Delta c - \frac{c}{\tau_S} + F - \Omega^{-1}\partial_t \Phi, && (4)\\ \alpha\partial_t \Phi &= \xi^2 \Delta\Phi + sin(2\pi(\Phi - \Phi_S)) + \lambda c(1 + cos(2\pi(\Phi - \Phi_S))), && (5) \end{aligned}$$

in the domain S, where α is the time relaxation parameter, ξ is the width of steps between terraces, Φ_S is the height of the initial substrate surface and λ is the coupling constant.

The boundary conditions are given by

$$\frac{\partial c}{\partial n}(t, x) = \frac{\partial \Phi}{\partial n}(t, x) = 0, t \in (0, T), x \in \partial S. \qquad (6)$$

The initial conditions are given by

$$\begin{aligned} c(0, x) &= 0, x \in S, && (7)\\ \Phi(0, x) &= \Phi_S(x), x \in S. && (8) \end{aligned}$$

3 Numerical scheme

We use an explicit scheme of the finite difference method to solve the free boundary problem of epitaxial crystal growth. The first step in the discretization is to divide the problem's domain into a two-dimensional grid and then derivatives are replaced with equivalent finite differences.

We consider the problem's domain S to be a rectangular domain $(0, L_1) \times (0, L_2)$ which is to be discretized. We partition the domain S using a grid of internal nodes $\omega_h = \{(ih_1, jh_2)|i = 1, ..., N_1 - 1, j = 1, ..., N_2 - 1\}$, where $h_1 = \frac{L_1}{N_1}, h_2 = \frac{L_2}{N_2}$ are the mesh sizes in S. We discretize the time interval using a mesh $[0, T] : T_\tau = \{k\tau|k = 0, ..., N_T\}$, where $\tau = \frac{T}{N_T}$ is a time step. Then we can consider a grid function $u : T_\tau \times \omega_h \to \mathbb{R}$ for which $u^k_{ij} = u(ih_1, jh_2, k\tau)$.

The time derivative is approximated by forward difference

$$\partial_t u^k_{ij} \approx \frac{u^{k+1}_{ij} - u^k_{ij}}{\tau},$$

and the space derivatives are approximated by second-order central differences:

$$\begin{aligned} \partial^2_x u^k_{ij} &\approx \frac{u^k_{i+1,j} - 2u^k_{ij} + u^k_{i-1,j}}{h_1^2},\\ \partial^2_y u^k_{ij} &\approx \frac{u^k_{i,j+1} - 2u^k_{ij} + u^k_{i,j-1}}{h_2^2}. \end{aligned}$$

Then the Laplace operator in two dimensions is given by $\Delta_h {u^k}_{ij} = \partial^2_x u^k_{ij} + \partial^2_y u^k_{ij}$.

The explicit scheme has the form

$$
\begin{aligned}
\alpha\frac{\Phi_{ij}^{k+1}-\Phi_{ij}^{k}}{\tau} &= \xi^2\Delta_h\Phi_{ij}^{k}+sin(2\pi(\Phi_{ij}^{k}-\Phi_{S_{ij}}^{k})) \\
&\quad +\lambda c_{ij}^{k}(1+cos(2\pi(\Phi_{ij}^{k}-\Phi_{S_{ij}}^{k}))) \qquad (9)\\
\frac{c_{ij}^{k+1}-c_{ij}^{k}}{\tau} &= D\Delta_h c_{ij}^{k}-\frac{c_{ij}^{k}}{\tau_S}+F-\Omega^{-1}\frac{\Phi_{ij}^{k+1}-\Phi_{ij}^{k}}{\tau} \qquad (10)
\end{aligned}
$$

for $i=1,...,N_1-1, j=1,...,N_2-1, k=0,...,N_T$.

Discretization of the epitaxial crystal growth problem leads to a system of equations

$$
\begin{aligned}
\Phi_{ij}^{k+1} &= \Phi_{ij}^{k}+\frac{\tau\xi^2}{\alpha}\frac{\Phi_{i+1,j}^{k}+\Phi_{i,j+1}^{k}-4\Phi_{ij}^{k}+\Phi_{i,j-1}^{k}+\Phi_{i-1,j}^{k}}{h^2}\\
&\quad +\frac{\tau}{\alpha}sin(2\pi(\Phi_{ij}^{k}-\Phi_{S_{ij}}^{k}))\\
&\quad +\frac{\tau\lambda}{\alpha}c_{ij}^{k}(1+cos(2\pi(\Phi_{ij}^{k}-\Phi_{S_{ij}}^{k}))) \qquad (11)\\
c_{ij}^{k+1} &= c_{ij}^{k}+\tau D\frac{c_{i+1,j}^{k}+c_{i,j+1}^{k}-4c_{ij}^{k}+c_{i,j-1}^{k}+c_{i-1,j}^{k}}{h^2}\\
&\quad -\frac{\tau}{\tau_S}c_{ij}^{k}+\tau F-\frac{\Phi_{ij}^{k+1}-\Phi_{ij}^{k}}{\Omega} \qquad (12)
\end{aligned}
$$

for $i=1,...,N_1-1, j=1,...,N_2-1, k=0,...,N_T$. That means we can obtain the values at time $k+1$ from the corresponding ones at time k.

For $h=h_1=h_2$ this explicit method is known to be numerically stable and convergent whenever $\frac{\xi^2\tau}{\alpha h^2}\leq\frac{1}{4}$ and $\tau(\frac{4D_S}{h^2}+\frac{1}{\tau_S})\leq 1$.

The boundary conditions are treated by mirroring the values in the inner nodes across the boundary.

4 Numerical results

In the numerical experiments we investigated the influence of the parameter τ_S to the spiral growth. First, transient dynamics is quantified by defining the so called surface width $w(t)$ which is the mean fluctuation of the surface height

$$w(t)=\frac{1}{2}\langle\Phi(x,t)^2-\langle\Phi(x,t)\rangle^2\rangle^{1/2},$$

where $\langle f\rangle=L^{-2}\int_S f dx$. $(L=h(N-1)=50)$ (see Fig. 3).

Next, the parameters are set up as follows: $\Omega=2.0$, $\alpha=1.0$, $\xi=1.0$, $\lambda=10.0$, $D_S=2.0$, $F=3.0$, $\tau=0.00025$, $N_T=100000$, so that $T=25$. The dimensions of ω_h are 100×100 and the spatial step size is set to 50/99. The initial height of the substrate Φ_S is formed by $\frac{arctan(y/x)}{2\pi}$ for the dislocation. We observed two distinguished growth regimes. As can be seen in Fig. 3 for small τ_S, the spiral finds its final step spacing l essentially after a single rotation. In contrast, for very large τ_S the transient spiral ridge

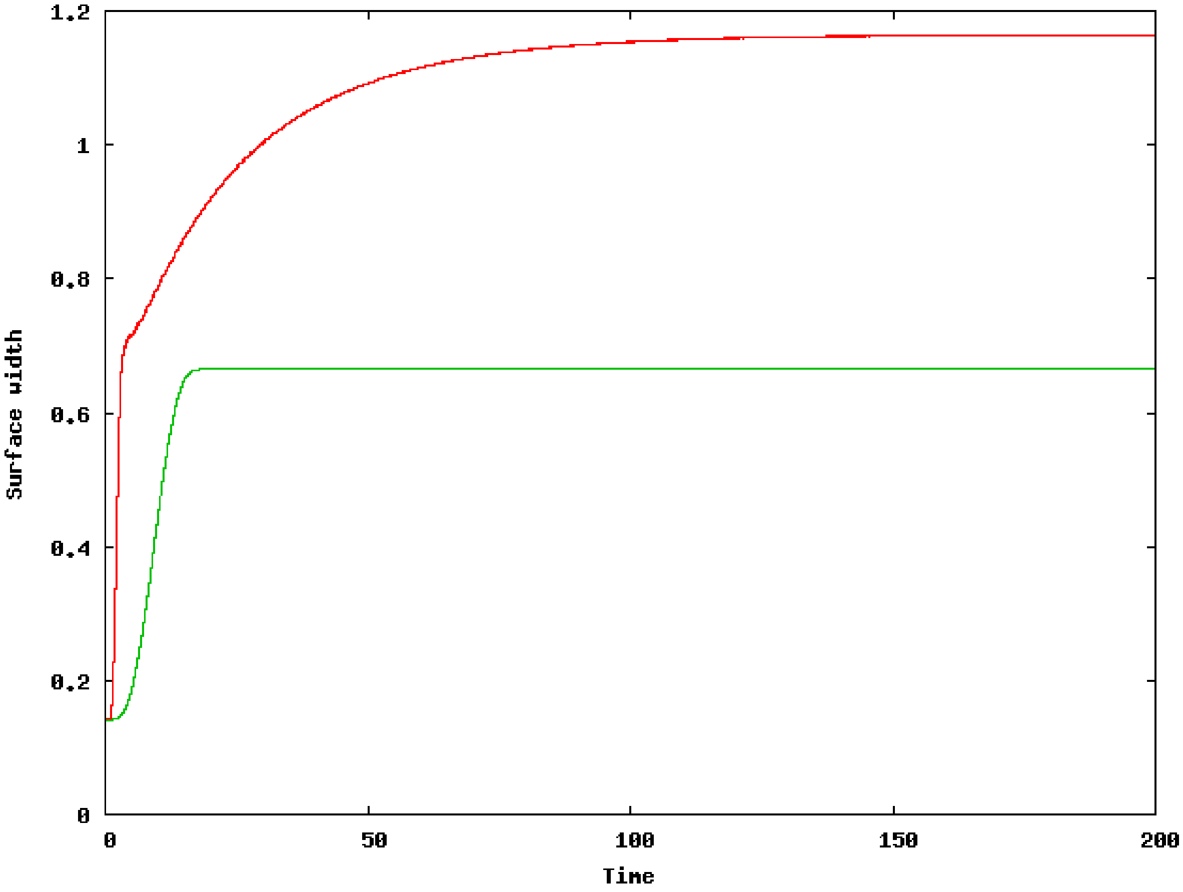
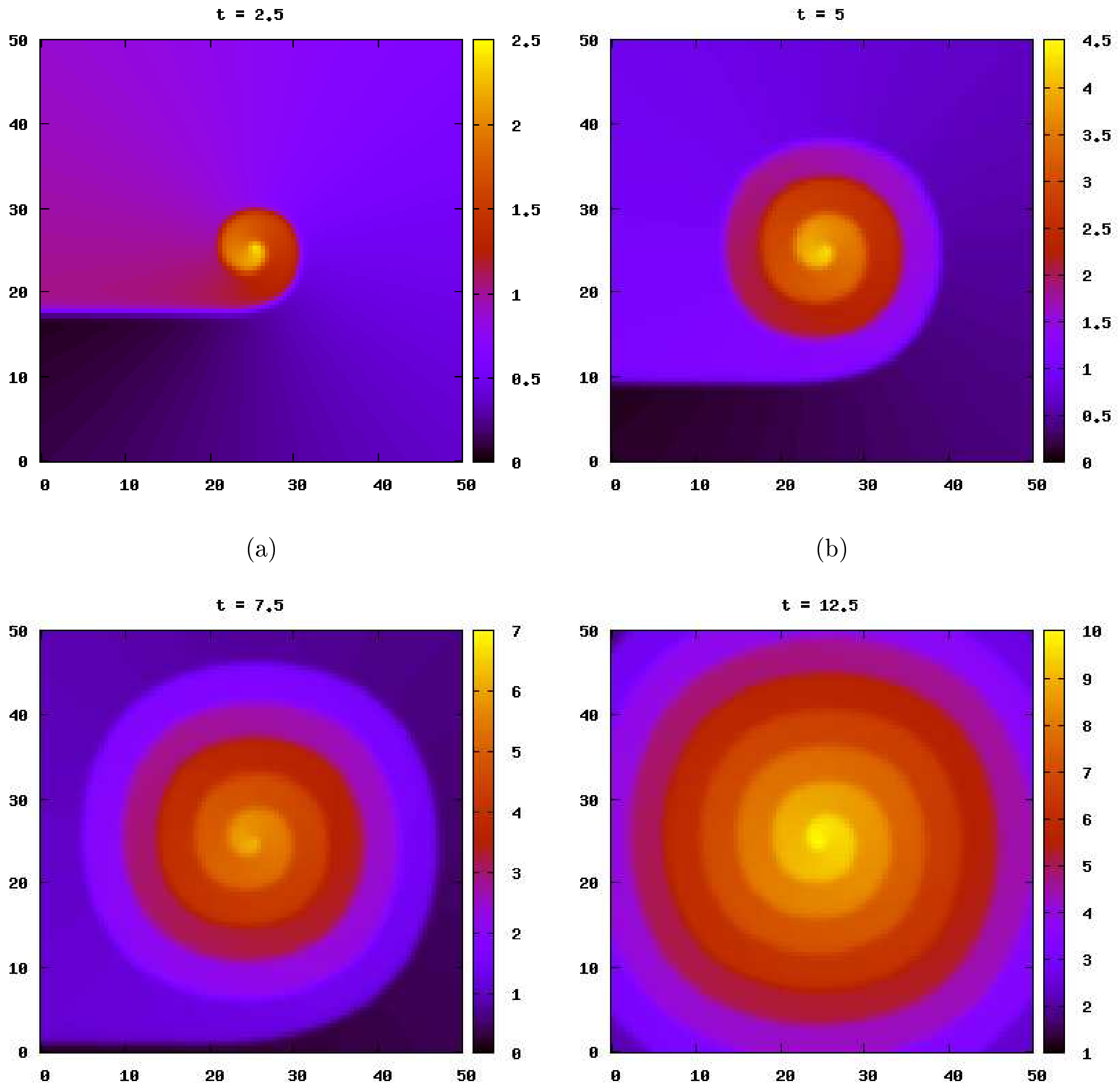


Figure 2: Comparison of transient dynamics for different desorption times. Green line: $\tau_S = 0.1$, the surface width quickly levels off and remains constant. Red line: $\tau_S = 10^{100}$, the surface width changes slowly in time.

evolves slowly towards a spiral with a constant l. This surface evolution is demonstrated in Fig. 4.

From these numerical simulations we conclude that step spacing is dependent on desorption time. The larger desorption time is, the smaller the step spacing is. Finally, we would consider elastic deformation of the solid generated by the misfit strain between atoms in the epitaxial layer and the substrate and include it to the model.



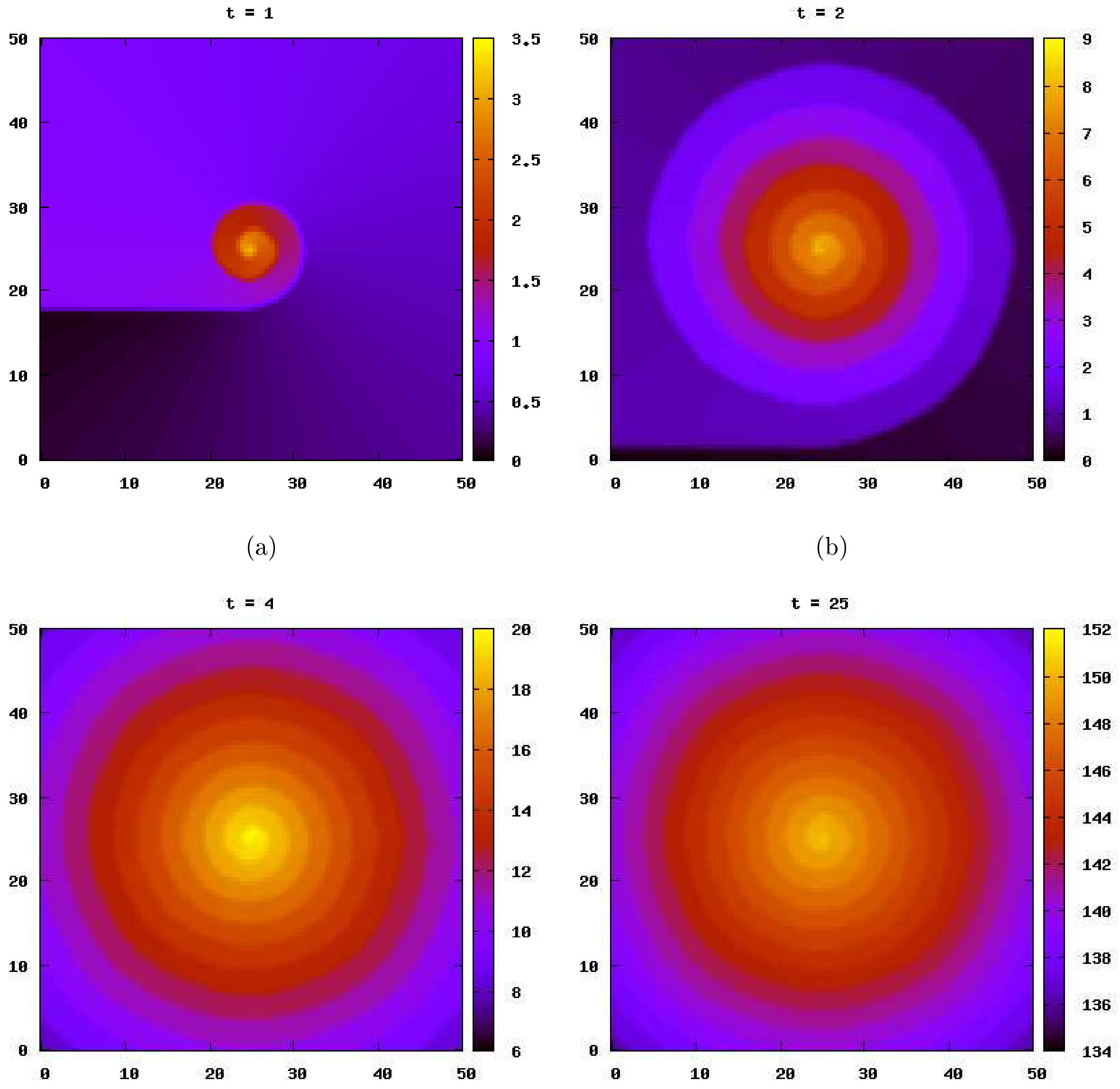
(a)

(b)

(c)

(d)

Figure 3: Spiral ridge at different times t for $\tau_S = 0.1$. Colour palette represents the surface height.



(a)

(b)

(c)

(d)

Figure 4: Spiral ridge at different times t for $\tau_S = 10^{100}$. Colour palette represents the surface height.

References

[1] W. K. Burton, N. Cabrera, and F. C. Frank, *The Growth of Crystals and the Equilibrium Structure of their Surfaces*, Phil. Trans. Roy. Soc. 243 (1951), 299.

[2] K. Byrappa, T. Ohachi, *Crystal Growth Technology*, William Andrew Publishing, 2002.

[3] Ch. Eck, H. Emmerich, *Liquid-phase epitaxy with elasticity*, Preprint 197, DFG SPP 1095 "Mehrskalenprobleme", 2004.

[4] H. Emmerich, *Modeling elastic effects in epitaxial growth*, Continuum Mech. Thermodyn., 15 (2003), pp. 197-215.

[5] J.-S. Guo, K. I. Nakamura, T. Ogiwara, J.-C. Tsai, *On the Steadily Rotating Spirals*, Japan J. Indust. Appl. Math., 23 (2006), pp. 1–19.

[6] V. Chalupecký, H. Emmerich, *Numerical scheme for two-scale model of liquid phase epitaxy*, In: Beneš M., Kimura M. and Nakaki T., Eds. *Proceedings of Czech Japanese Seminar in Applied Mathematics 2006*, in COE Lecture Note, Vol. 6, Faculty of Mathematics, Kyushu University Fukuoka, 2007, ISSN 1881-4042, pp. 50–61.

[7] A. Karma, and M. Plapp, *Spiral Surface Growth without Desorption*, Phys. Rev. Lett. 81 no. 20 (1998), pp. 4444–4447.

[8] F. Liu, and H. Metiu, *Stability and Kinetics of Step Motion on Crystal Surfaces*, Phys. Rev. E 49 (1997), pp. 2601–2616.

[9] T. Ohtsuka, *Motion of Interfaces by an Allen–Cahn Type Equation with Multiple-well Potentials*, Asymptotic Analysis, 56 (2008), pp. 87–123.

[10] F. Otto, P. Penzler, T. Rump, *Discretisation and Numerical Tests of a Diffuse–Interface Model with Ehrlich–Schwoebel Barrier*, In: em Multiscale modeling in epitaxial growth, vol. 149 of Internat. Ser. Numer. Math. Birkhauser, Basel, 2005, pp. 127–158.

[11] M. H. Sadd, *Elasticity: Theory, Applications and Numerics*, Academic Press, 2004.

[12] I. Sunagawa, *Crystals: Growth, Morphology and Perfection*, Cambridge University Press, 2005.

Exploitation of Particle Filter in Early Phase of a Reactor Accident[*]

Radek Hofman

3rd year of PGS, email: hofman@utia.cas.cz
Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, CTU in Prague
advisor: Ing. Petr Pecha, CSc., Institute of Information Theory and Automation, ASCR

Abstract. Exploitation of the data assimilation methodology in the field of radiation protection is studied. When radioactive pollutants are released into the atmosphere, a radioactive plume is passing over the terrain. In order to ensure efficiency of introduced countermeasures, it is necessary to predict spatial and temporal distribution of the aerial pollution and material already deposited on the ground. The predictions are made by the means of numerical dispersion models with many inputs. A group of the most significant input parameters affecting the dispersion process was selected using available sensitivity and uncertainty studies performed on dispersion models. Exact values of these parameters are uncertain due to the stochastic nature of atmospheric dispersion, hence the parameters are modeled as random quantities. Data assimilation is the optimal way how to exploit information from both the measured data and expert-selected prior knowledge to obtain reliable estimates of the input parameters. Early identification of the parameters is essential for reduction of uncertainty of the radiation situation predictions. In this paper, sampling-importance-resampling algorithm (particle filter) is used to evaluate posterior distribution of estimated parameters and improve their estimates on-line as the plume is passing over the stationary measuring sites. The algorithm is tested on an artificial release scenario.

Abstrakt. Příspěvek pojednává o využití data asimilace v časné fázi radiační nehody. V případě vzdušného úniku radionuklidů se utvoří mrak postupující nad terénem. V rámci zajištění ochrany obyvatelstva formou vhodných protiopatření je nutná znalost předpovědi časového a prostorového rozložení radionuklidů. Predikce se počítají pomocí numerických disperzních modelů s mnoha vstupy. Na základě studie citlivosti a neurčitosti provedené na disperzních modelech byla vybrána podmnožina nejdůležitějších vstupních parametrů. Jejich přesná hodnota je kvůli stochastické povaze atmosférické disperze neznámá a tak jsou tyto parametry modelovány jako náhodné veličiny. Data asimilace je optimální způsob využití expertně volené apriorní informace a dostupných meřených dat k získání zpřesněných odhadů vstupních parametrů modelu. Jejich včasná identifikace je stěžejní pro redukci neurčitosti v předpovědích. V příspěvku je demonstrováno využití sampling-importance-resampling algoritmu pro odhad posteriorní distribuce odhadovaných parametrů a vylepšovnání jejich odhadu on-line v průběhu postupu mraku nad terénem. Algoritmus je testován na scénáři simulovaného úniku.

1 Introduction

During the operation of a nuclear power plant, there is a potential for accidental release and dispersion of a nuclear material into ambient atmosphere and exposure of population

[*] This work has been supported by the grant GACR No. 102/07/1596 and the MSTM project 1M0572.

to the ionizing radiation. The radiation dose received by the public as a consequence of a release comes mostly from five sources: External γ-radiation from the plume (cloud shine); external γ-radiation from radioactive material deposited on the ground, trees, buildings (ground shine); inhalation of radioactive material; external α, β and γ from radioactive material deposited on the skin and ingestion of contaminated foodstuff.

The time lapse of a nuclear release can be split into two major phases. The first phase, the early phase, covers the first few hours or days and lasts until the radioactive cloud has passed the area of interest. During this phase, the irradiation from cloud shine, ground shine, skin contamination and inhalation are most important. The second phase, the late phase, lasts until the radiation levels resumes back to levels of background. In this phase, dose from ground shine and ingestion becomes important. Negative impacts on population health are averted by the means of countermeasures introduced as soon as possible after or even before the expected release. These can be iodine prophylaxis, food bans, sheltering or evacuation.

The unavoidable condition for application of effective countermeasures is knowledge of spatial and temporal distribution of radioactive pollutants. Former accidents on nuclear facilities revealed unsatisfactory level of the decision support, both in hardware equipment (reliable communication channels, computation techniques) and also deficiencies in software decision support systems (DSS). Great attention to this topic is paid since the Chernobyl disaster. DSS is a software tool including a mathematical model for prediction of radionuclide spreading in the environment [9]. It can embody different subsystems for evaluation of expected consequences in terms of demographic or economic statistics. Output from the system should provide to responsible authorities a rational basis for coordination of countermeasures [11], [7].

Data assimilation is a way how to increase reliability of such predictions in both the early and the late phase of an accident [13]. Recent development in hardware allows us to implement assimilation algorithms based on methods earlier computationally prohibited like sequential Monte Carlo methods [3]. Marginalized particle filter [12] was used here to estimate model error covariance structure in a parametrized form. Data assimilation is the optimal way how to exploit information from both the measured data and expert-selected prior knowledge to obtain reliable estimates. This paper studies exploitation of the data assimilation in the early phase of an accident when the radioactive cloud is passing over the terrain.

The outline of this paper is as follows. Problem statement is given in Section 2. Atmospheric dispersion model and methodology of calculation of cloud shine dose are described here. Section 3 briefly discusses particle filter and puts it in the scope of the Bayesian filtering. Section 4 presents a particular assimilation scenario and numerical experiment with simulated measurements. Conclusion and future work is given in Section 5.

2 Problem statement

Assume an accident in a nuclear power plant followed by aerial release of radionuclides. After the release, there is a radioactive cloud passing over the terrain. The spatio-temporal distribution of radionuclides is modeled by the means of numerical dispersion models in order to determine appropriate countermeasures. Output of such a model is a prediction of radiation situation given in terms of radiological quantities. Assume that the radiological quantity of interest is the continuous activity concentration in air $C(\boldsymbol{s},\, t)$, where $\boldsymbol{s} = (s_1,\, s_2,\, s_3)$ is a vector of spatial coordinates and $t = 1,\, \ldots,\, t_{\mathrm{MAX}}$ is the time index. Concentration of activity is important radiological quantity which can be used for calculation of some other quantities like deposition or doses from different pathways of irradiation. The concentration itself is a difficult quantity to measure, therefore the measuring devices are designed to measure the γ-dose rate. It has well developed measuring methodology. These measurements can be provided by stationary measuring sites or mobile groups [10].

For computational reasons, the continuous quantity $C(\boldsymbol{s},\, t)$ is evaluated only in a set of M points of a computational grid in time t. Values of $C(\boldsymbol{s}, t)$ in the grid points are aggregated in vector $\boldsymbol{C}_t$. The available measurements of time integrated γ-dose rate at time t are aggregated in vector $\boldsymbol{y}_t$. We can employ data assimilation and use the sparse measurements to improve reliability of model predictions and thus allow for introduction of effective countermeasures in the actually affected areas.

The evolution of $C(\boldsymbol{s},\, t)$ is modeled by a dispersion model which is parametrized by a set of parameters $\boldsymbol{\Theta}_t$. These parameters reflect physical processes involved in the atmospheric dispersion, atmospheric conditions and conditions of the accident in each time step t. Exact values of the parameters are uncertain due to stochastic nature of the dispersion, lack of accurate information, etc. Typically, the choice of values of these parameters is subject to an expert opinion. The subjective choice of parameter values can introduce significant errors into the predictions. To avoid this, we apply Bayesian approach and treat the parameters as random quantities. We attempt to estimate parameter distributions in consecutive time step from measurements. The number of parameters is potentially large but a restricted subset $\boldsymbol{\theta}_t \subset \boldsymbol{\Theta}_t$ of the most important parameters can be found for specific scenario [8].

Since all uncertainty is modeled by probability distributions, the appropriate data assimilation methodology is the Bayesian filtering. The introduced scenario fits into the family of state-space models. Realization of the process at time t contains all the information about the past, which is necessary in order to calculate the prediction of future evolution. State vector $\boldsymbol{x}_t$ of the system comprises of the two components $\boldsymbol{x}_t = [\boldsymbol{C}_t, \boldsymbol{\theta}_t]^T$. The model of integrated γ-dose rate measurements $\boldsymbol{y}_t$ is given by the probability density function (pdf) $p(\boldsymbol{y}_t|\boldsymbol{x}_t)$.

2.1 Evolution of state

Evolution of the state is given by the transition pdf $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$:

$$\begin{aligned} p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) &= p(\boldsymbol{C}_t,\, \boldsymbol{\theta}_t|\boldsymbol{C}_{t-1},\, \boldsymbol{\theta}_{t-1}) \\ &= p(\boldsymbol{C}_t|\boldsymbol{C}_{t-1},\, \boldsymbol{\theta}_t,\, \boldsymbol{\theta}_{t-1})p(\boldsymbol{\theta}_t|\boldsymbol{C}_{t-1},\, \boldsymbol{\theta}_{t-1}) \end{aligned} \tag{1}$$

Under the choice of atmospheric dispersion model $C_{\mathrm{ADM}}(\boldsymbol{\theta}_t)$ and its parameters $\boldsymbol{\theta}_t$, the evaluation of $\boldsymbol{C}_t$ is deterministic:

$$p(\boldsymbol{C}_t|\boldsymbol{C}_{t-1},\, \boldsymbol{\theta}_t,\, \boldsymbol{\theta}_{t-1}) = \delta(\boldsymbol{C}_t - C_{\mathrm{ADM}}(\boldsymbol{\theta}_t)) \tag{2}$$

Time evolution of $\boldsymbol{\theta}_t$ is given by the pdf $p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})$. Under the choice of time invariant parameters ($\boldsymbol{\theta}_t = \boldsymbol{\theta}$), the transition pdf gets the form $p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}) = \delta(\boldsymbol{\theta}_t - \boldsymbol{\theta})$. The process is initialized with prior pdf $p(\boldsymbol{\theta}_0)$, typically covering wide range of possibilities.

We chose the Gaussian puff model (GPM) for the atmospheric dispersion model. It is based on approximative solution of the three dimensional advection-diffusion equation [1]:

$$C(\boldsymbol{s},\, t) = \frac{Q\, f_{\mathrm{D}}(t)\, R(t)}{(2\pi)^{\frac{3}{2}} \sigma_{s_1}\, \sigma_{s_2}\, \sigma_{s_3}} \exp\left\{ -\frac{1}{2} \left[\left(\frac{s_1 - ut}{\sigma_{s_1}} \right)^2 + \left(\frac{s_2}{\sigma_{s_2}} \right)^2 + \left(\frac{s_3}{\sigma_{s_3}} \right)^2 \right] \right\}, \tag{3}$$

where t is time index, Q is the total released activity in Bq and u is the wind speed. Dispersion coefficients $\{\sigma_{s_i}\}|_{i=1,2,3}$ are functions of distance from the source. Factor $f_{\mathrm{D}}(t)$ stands for radioactive decay, dry and wet deposition. The last term $R(t)$ accounts for homogenization of the vertical profile of concentration due to the reflections from the top of mixing layer and the ground. See [4] for more details.

2.2 Measurement model

Measurements are assumed to be normally distributed and mutually independent given the state $\boldsymbol{x}_t$. Errors of measurements are set proportional to the their values with an offset term modeling the background radiation superposed to the actual dose measurements

$$\boldsymbol{y}_t \sim \mathcal{N}(\boldsymbol{D}_t,\, \boldsymbol{\Sigma}(\boldsymbol{D}_t)), \tag{4}$$

where $\mathcal{N}(\boldsymbol{a},\, \boldsymbol{\Sigma})$ is a multidimensional normal distribution with mean value $\boldsymbol{a}$ and a covariance matrix $\boldsymbol{\Sigma}$. $\boldsymbol{D}_t$ is a vector of measurements of time integrated absorbed γ-dose in all the measuring sites available in time t. If the released nuclide is a noble gas, there is no deposition and we don't have to assume ground shine from deposited material. In this case, the measured quantity is just the γ-dose from cloud shine. The time integral of absorbed γ-dose rate in tissue from a mixture of radionuclides emitting photons on different energy levels $E_{\gamma,j}$is

$$D_{i,t} = \int\limits_{t-1}^{t} \sum_j \frac{K_j\, \mu_{a,j}\, E_{\gamma,j}}{\rho}\, \Phi_j(C(\boldsymbol{s}_{(i)}, \tau))\, d\tau, \tag{5}$$

where K_j, $\mu_{a,j}$ and Φ_j are conversion coefficient, absorption coefficient and effective flux of gamma rays, respectively. Subscript j stands for the fact, that the particular values depend on the energy level $E_{\gamma,j}$. Summation is over assumed energy levels and ρ is the mass density of air. Equation (5) defines the observation operator converting the concentration in $Bq\, m^{-3}$ to the time integrated γ-dose in Gy.

The general expression for Φ at a receptor located at $\tilde{\boldsymbol{s}} = (\tilde{s}_1,\, \tilde{s}_2,\, \tilde{s}_3)$ from a source of energy E_γ dispersed in air is

$$\Phi(\tilde{s}_1,\, \tilde{s}_2,\, \tilde{s}_3,\, E_\gamma) = \iiint \frac{f(E_\gamma) B(E_\gamma, \mu r) C(s_1,\, s_2,\, s_3)}{4\pi\, r^2}\, ds_1\, ds_2\, ds_3, \tag{6}$$

where $r^2 = (\tilde{s}_1 - s_1)^2 + (\tilde{s}_2 - s_2)^2 + (\tilde{s}_3 - s_3)^2$, $f(E_\gamma)$ is the branching ratio to the specific energy, μ is the attenuation coefficient of air, $B(E_\gamma, \mu r)$ is the dose build-up factor, $C(\boldsymbol{s})$ is the radionuclide concentration in $Bq\, m^{-3}$ of isotope being considered. The build-up factor can be calculated from Bergers analytical expression

$$B(E_\gamma, \mu\, r) = 1 + a\, \mu r\, \exp(b\, \mu r), \tag{7}$$

where coefficients μ, a and b depend on E_γ. Energy dependent absorption coefficient μ_a is calculated as

$$\mu_a = \mu / \left[1 + \frac{a}{(1-b)^2} \right]. \tag{8}$$

The simplicity of used Gaussian puff model (3) allows for numerical evaluation of integral (6) on a compact support where the concentration is not negligible. If the radioactive plume is large compared to the mean free path of the γ-rays, then the semi-infinite cloud approximation of effective flux can be successfully used. See [5] for more details.

3 Data assimilation

Bayesian approach to data assimilation is based on representing uncertainty in the state via probability distribution. When no measurements are available the probability distribution of the considered state (the prior) must be rather wide to cover all possible realizations of the state. Each incoming measurement brings information about the “true” state, reducing the original uncertainty. In effect, with increasing number of measurements, the posterior pdf is narrowing down around the best possible estimate.

Formally, the prior distribution $p(\boldsymbol{x}_0)$ is transformed into posterior pdf $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ using measurements $\boldsymbol{y}_{1:t} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t\}$ by recursive repetition of the following steps:

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}) = \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1}) d\boldsymbol{x}_{t-1} \tag{9}$$

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t) p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})}{\int p(\boldsymbol{y}_t|\boldsymbol{x}_t) p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}) d\boldsymbol{x}_t}, \tag{10}$$

The process is initialized by prior $p(\boldsymbol{x}_0)$.

Evaluation of (9) and (10) involves integration over complex spaces and often it is computationally infeasible. Suboptimal solution can be found by the means of sequential Monte Carlo methods also known as particle filters [2]. Particle filters numerically approximate posterior pdf $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ using a set of particles $\boldsymbol{x}_t^{(i)}$ and importance weights $w_t^{(i)}$ for $i = 1, 2, \ldots, N$:

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)}\, \delta(\boldsymbol{x}_t - \boldsymbol{x}_t^{(i)}), \tag{11}$$

where $\delta()$ is the Dirac δ-function. The particles $\boldsymbol{x}_t^{(i)}$ are drawn from a proposal pdf $q(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$, which can be an arbitrary pdf the support of which includes the support of

$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$. Under this approximation, the integral equations (9)–(10) reduces to drawing new particles at each time t and simple re-evaluation of the importance weights:

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(i)})p(\boldsymbol{x}_t^{(i)}|\boldsymbol{x}_{t-1}^{(i)})}{q(\boldsymbol{x}_t^{(i)}|\boldsymbol{x}_{t-1}^{(i)}, \boldsymbol{y}_{1:t})}. \tag{12}$$

Here, $\propto$ denotes equality up to multiplicative constant. This constant can be easily computed to assure that $\sum_{i=1}^{N} w_t^{(i)} = 1$. Equation (12) can be further simplified to $w_t^{(i)} \propto w_{t-1}^{(i)} p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(i)})$ by choosing $q(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) \equiv p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$.

The approximation is easily extendable for prediction. Predicted pdf of the state at time $t+k$ is then approximated as

$$p(\boldsymbol{x}_{t+k}|\boldsymbol{y}_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)}\, \delta(\boldsymbol{x}_{t+k} - \boldsymbol{x}_{t+k}^{(i)}), \tag{13}$$

where particles $\boldsymbol{x}_{t+k}^{(i)}$ are recursively generated from $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{(i)})$.

4 Numerical experiment

For purposes of numerical experiment was chosen assimilation scenario with an instantaneous release of ^{41}Ar. Numerical experiment is conducted as a twin experiment, where the measurements are simulated via a twin model and perturbed. Convergence of radiological quantity of interest—^{41}Ar activity concentration in air—evaluated on basis of estimated parameters to that produced by the twin model can be then assessed.

Since the argon is a noble gas, there is no deposition and consequently no ground shine. The released activity is propagated via Gaussian puff model, (3). Half life of decay of ^{41}Ar is 109.34 minutes. According to the TORI (Tables Of Radioactive Isotopes) database, there are more energy levels of γ radiation produced by isotope ^{41}Ar. We assume just the energy level 1293.57keV with the branching ratio 99.1%. The rest being included in the 0.9% is neglected and the summation over energy levels in (5) can be omited.

The topology of measuring sites is similar to that of the Early Warning Network of the Czech Republic [10]. The source of simulated release is a nuclear power plant surrounded by almost fifty stationary measuring sites capable to measure time integrated γ-dose (5). Measuring sites are located more or less regularly in the area of radius 10km around the source. The time step of assimilation algorithm was set to 10 minutes and the time horizon t_{MAX}=6 (60min). Measuring devices are assumed to integrate the γ-dose in 10 minute intervals and then send measurements on-line to the quarters of crisis management. The height or release is 50m and the magnitude of release Q=1.0E+10Bq of ^{41}Ar. We assume no vertical velocity or any significant heat capacity of the effluent and the effective height remains 50m during the puff propagation. The time horizon spans up to 1 hour after the release start. It means, that we performed 6 assimilation cycles consisting of time and data update steps.

4.1 Parametrization of atmospheric dispersion model

A group of the most significant variables affecting the dispersion process (including meteorological inputs) was selected using available sensitivity and uncertainty studies performed on Gaussian dispersion models [8]. Variables of the dispersion model C_{ADM} treated in this numerical example as uncertain are: magnitude of release Q, horizontal dispersion coefficients $\sigma_{s_i}|_{i=1,2}$ and also two meteorological inputs: wind speed u and wind direction ϕ. Their parametrization via vector of random parameters $\boldsymbol{\theta}_t = (\omega_t,\, \xi_t,\, \psi_t,\, \zeta_t)$ and location parameters $(Q_0,\, u_0,\, \phi_0, \sigma_{s_{i0}}|_{i=1,2})$ is listed in Table 1. The parametrization was selected according to that in the UFOMOD code [6]. Location parameters refer to the prior initialization of the variables. All the random parameters are treated as time constant: $\boldsymbol{\theta}_t = \boldsymbol{\theta}$, even the parameters ξ_t and ψ_t concerning uncertainty in meteorological forecast. In case of time horizon of several hours, the assumption of stationarity of the meteorological condition vanishes. Parametrization of the meteorological data has to be fragmented into shorter time intervals (usually hourly intervals) where the assumption of stationarity holds.

variable	**physical effect**	**parametrization**
Q	magnitude of release	$Q = \omega_t\, Q_0$
u	wind speed	$u = (1 + 0.1\, \xi_t)\, u_0 + 0.5\, \xi_t$
ϕ	wind direction	$\phi = \phi_0 + \Delta\phi,\ \Delta\phi = \psi_t\, (2\pi/80)$ rad.
$\sigma_{s_i}\|_{i=1,2}$	horizontal dispersion	$\sigma_{s_i} = \zeta_t\, \sigma_{s_{i0}}\|_{i=1,2}$

Table 1: Parametrization of selected variables and inputs to the ADM.

The set of parameters $\boldsymbol{\theta}_{\mathrm{TWIN}}$ used for evaluation of the twin model simulating measurements is

$$\boldsymbol{\theta}_{\mathrm{TWIN}} = (0.72,\ -0.17,\ -8.3,\ 1.3). \tag{14}$$

The comparison of initial C_{ADM} inputs with the initial setting and the twin model is in Table 2. The real release was smaller in magnitude, with the lower wind speed, directed by approximately 37deg anticlockwise and the puff dispersed more than was apriori assumed. Horizontal dispersion parameters σ_{s_1} and σ_{s_2} are functions of distance the from source. The total number of $N = 1000$ particles was initialized with random vectors $\{\boldsymbol{\theta}^{(i)},\, i = 1, \ldots, 1000\}$ with elements generated according to the pdfs in Table 3.

variable	**physical effect**	**prior val.**	**param. value**	**true value**
Q	released activity	1.0E+10Bq	0.72	7.2E+09Bq
u	wind speed	3.10m/s	-0.17	2.96m/s
ϕ	wind direction	310.0deg	-8.3	272.7deg
$\sigma_{s_i}\|_{i=1,2}$	horizontal disp.	$\sigma_{s_i} = \sigma_{s_i}(dist)\|_{i=1,2}$	1.3	$\sigma_{s_i} = 1.3\, \sigma_{s_i}\|_{i=1,2}$

Table 2: Values of variables of the initial model setting and the twin model.

parameter	physical effect	pdf type	mean value	std. dev.
ω_t	magnitude of release	log-normal	1.0	1.0 (3σ truncated)
ξ_t	wind speed	uniform	0.0	1.0
ψ_t	wind direction	uniform	0.0	10.0
ζ_t	horizontal dispersion	log-normal	1.0	1.0 (3σ truncated)

Table 3: Prior distributions of estimated parameters $\boldsymbol{\theta}_t = (\omega_t,\, \xi_t,\, \psi_t,\, \zeta_t)$.

4.2 Results

The results are visualized in terms of the time integral of ground level concentration of activity in air (TIC):

$$TIC(\boldsymbol{s}) = \int\limits_{0}^{t_{\mathrm{MAX}}} C(\boldsymbol{s},\, \tau)\, d\tau. \tag{15}$$

Computational grid is a rectangular grid of dimension 41 × 41 grid points with the grid step $1km$. The source of pollution is placed in the center of the grid.

In Figure 1 left we can see the TIC evaluated by the atmospheric dispersion model without the data assimilation and with initial setting of variables $Q = Q_0$, $u = u_0$, $\phi = \phi_0$ and $\sigma_{s_i}|_{i=1,2} = \sigma_{s_{i0}}|_{i=1,2}$. This is done by setting $\boldsymbol{\theta} = (1.0, 0.0, 0.0, 1.0,)$, see Table 1. In Figure 1 right is the TIC evaluated by the twin model used for simulation of measurements. In Figure 2 are visualized assimilation results. Assimilation results are presented in the form of expected value of TIC with respect to the predictive densities at different time steps. Expected value of prediction of TIC displayed in Figure 2 top left are based only on the measurements $\boldsymbol{y}_1$. Even at this stage, the wind direction was correctly recognized, however other parameters, such as parametrization of Q, are still too uncertain and the prediction differs from the twin model. With increasing time the measurements provide enough information and the expected values of TIC are converging to the twin model.

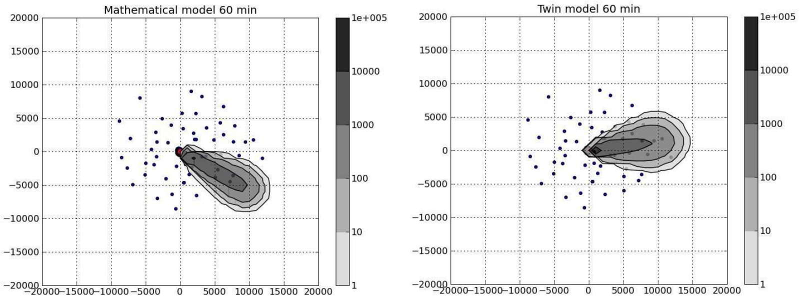


Figure 1: Predicted TIC based on initial values without the data assimilation (left) and the twin model (right).

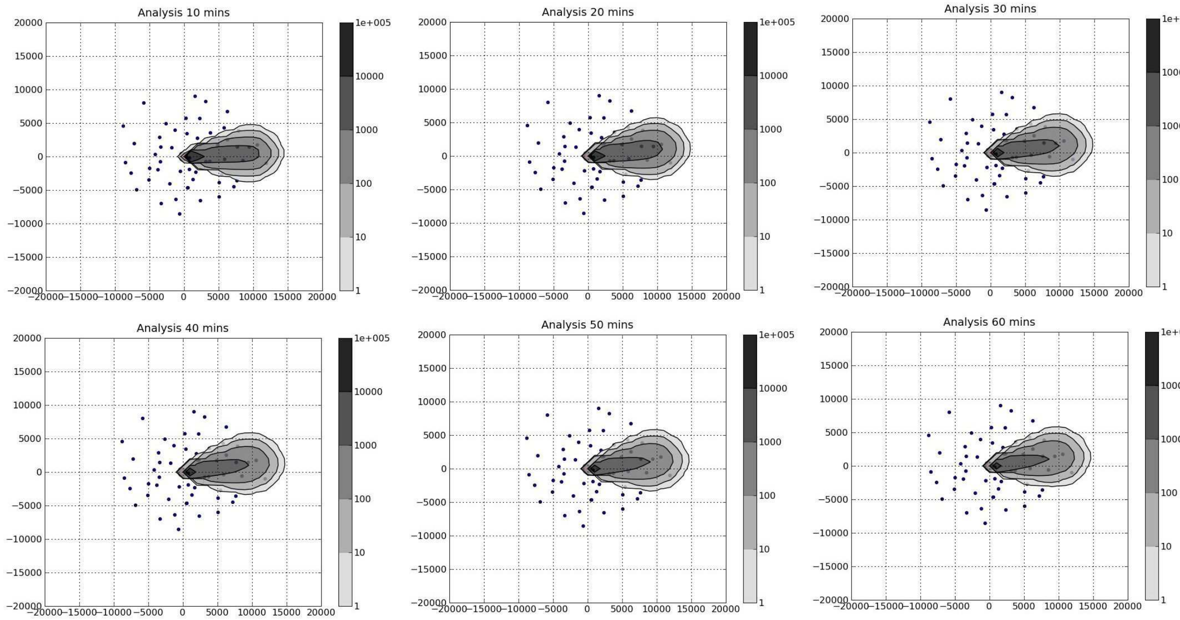


Figure 2: Predicted TIC based on assimilation at $t = 1, 2, 3, 4, 5, 6$, respectively.

5 Conclusion

Rapid assessment of the situation in case of an aerial release of radionuclides is crucial for planning of countermeasures. Introduced Bayesian methodology has very interesting properties suitable for this scenario. Specifically, it allows joint estimation of spatio-temporal distribution of activity and parameters of the dispersion model. Thus, we obtain assimilated estimate of the radiation situation on the terrain and a way how to easily extend this estimates to predictions on an arbitrary horizon. The presented scenario clearly illustrates the power of the method. However, a lot of work is required to incorporate the method to the existing decision support systems. We foresee the core of the work in development of more realistic models of the state evolution and the measurements. For example, more realistic scenarios should consider a mixture of radionuclides and extended set of uncertain variables. Such extension of the model inevitably increases complexity of the implied algorithm which may lead to computational difficulties. These may be overcome with exploitation of recent developments in the filed of sequential sampling, such as adaptive resampling or problem specific proposal densities.

Acknowledgement

This work is part of the grant project GAČR No. 102/07/1596. This work is also supported by projects MŠMT 1M0572.

References

[1] R. Barrat. *Atmospheric dispersion modelling.* Earthscan, (2001).

[2] A. Doucet et al. *Sequential Monte Carlo methods in practice.* Springer Verlag, (2001).

[3] R. Hofman and P. Pecha. Data assimilation of model predictions of long-time evolution of Cs-137 deposition on terrain. In '2008 IEEE International Geoscience & Remote Sensing Symposium', (2008). Boston, Massachusetts, U.S.A.

[4] R. Hofman et al. A simplified approach for solution of time update problem during toxic waste plume spreading in atmosphere. In '10-th In. Conf. HARMO12, Cavtat, HR, October 6-10, 2008', (2008).

[5] T. J. Overcamp and R. A. Fjeld. A simple approximation for estimating centerline gamma absorbed dose rates due to a continuous gaussian plume, (1987).

[6] H. Panitz et al. UFOMOD - Atmospheric dispersion and deposition, (1989).

[7] P. Pecha and R. Hofman. Integration of data assimilation subsystem into environmental model of harmful substances propagation. In '11-th In. Conf. HARMO11, Cambridge, UK, July 2-5, 2007', (2007).

[8] P. Pecha and L. Housa. *Models of pollution propagation through the living environment - from deterministic to probabilistic estimation.* Safety of Nuclear Energy (journal of the Czech Nuclear Society), 2007, No. 1/2, pages 115/127 (2007).

[9] P. Pecha et al. Training simulator for analysis of environmental consequences of accidental radioactivity releases. In '6th EUROSIM Congress on Modelling and Simulation, Ljubljana, Slovenia', (2007).

[10] P. Pecha et al. Assimilation techniques in consequence assessment of accidental radioactivity releases. ECORAD 2008, Bergen, Norway, (2008).

[11] C. Rojas-Palma. Data assimilation for off site nuclear emergency management. Technical report, SCK-CEN, DAONEM final report, RODOS(RA5)-RE(04)-01, (2005).

[12] T. B. Schön et al. *Marginalized particle filter for mixed linear/nonlinear state-space models.* ESAIM: PROCEEDINGS **19** (2007), 53–64.

[13] J. Smith and S. French. Bayesian updating of atmospheric dispersion model for use after an accidental release of radioactivity, (1993).

Vztažení provenance pro ontologii k signatuře

František Jahoda

1. ročník PGS, email: jahoda@cs.cas.cz
Katedra matematiky, Fakulta jaderná a fyzikálně inženýrská, ČVUT v Praze
školitel: Július Štuller, Institute of Computer Science, ASCR

Abstract. In the Semantic web paradigm data are described by ontologies. The ontologies may be subject of changes. The paper deals with storing information about changes and querying stored information. Specifically, it propose a method for binding information about changes to a signature (set of symbols from the ontology).

Abstrakt. V kontextu Sémantického webu jsou data popsána ontologiemi. Tyto ontologie mohou podléhat změnám. Článek se zaobírá způsobem jak uchovávat informace o těchto změnách a jak je dotazovat. Konkrétněji navrhuje postup, jak vztáhnout informace o změnách k množině symbolům z ontologie (signatuře).

1 Úvod

Množství informací na webu se stále zvětšuje. Již dávno se není možné na webu orientovat bez pomoci specializovaných aplikací jako jsou vyhledávače. Nejčastěji používanými nástroji pro orientaci na webu, fulltextové vyhledávače, indexují obsah podle slov obsažených v dokumentech na webu. Tyto specializované nástroje mají problém rozlišit různou sémantiku slov vyplývající např. z kontextu. Při zvětšujícím se množství informací se méně obvyklé významy vyhledávají hůře. Integrace informací z více zdrojů ještě více zdůrazňuje chyby při určování sémantiky dat na webu. Množství dat na webu velmi rychle roste a stejně tak se na web dostávají nové druhy informací, proto se v budoucnu bude problém s určováním sémantiky dat a integrací dat zvětšovat.

Již dnes však existuje řešení, které je schopno výše popsané problémy řešit současnými prostředky. Toto řešení se nazývá Sémantický web [1] a spočívá v poskytování dat v přesně daném strojově čitelném formátu. Samozřejmě formát dat je určen pouze rámcově, aby mohl pokrýt většinu informací, jež se na webu vyskytují a vyskytovat budou. Přesnější určení formátu dat se provádí pomocí jazyka RDFS nebo OWL a konkrétní zadání dat pak pomocí jazyka RDF. Aplikace postavené na technologiích sémantického webu budou schopny provádět daleko lepší hledání a integraci informací dostupných na webu. Sémantický web tedy poskytuje již dnes dostupné řešení, jak překonat výše zmíněné problémy webu klasického.

Data na sémantickém webu lze popsat za pomoci ontologií [2] vyjádřených jazykem OWL. Ontologie umožňují definovat vztahy v datech a zároveň z existujících vztahů vytvářet vztahy nové. Ontologie neslouží jen k definici možných vztahů v datech, ale hlavně k integraci dat popsanými pomocí společné ontologie. V prostředí sémantického webu je tedy možné integrovat data z rozličných oblastí, pokud jsou popsány pomocí propojených ontologií. Způsobem jak vytvořit společnou ontologii určenou k integraci dat se zaobírá proces integrace ontologií. Protože integrace ontologií je poměrně náročný proces často s nutnou účastí doménového experta, je snaha se mu vyhnout používáním relativně malého množství ontologií. Tyto ontologie by měly být vytvořeny odborníky v oboru (doménovými experty) a používány pro velkou šíři aplikací. Vývojáři konkrétní aplikace často nemají dostatek znalostí k úpravám ontologie a musí se spoléhat na správce ontologie.

Existují oblasti, které lze přesně popsat ontologiemi, které nepodléhají změnám např. genealogie. Ovšem množství oblastí se vyvíjí a jejich ontologie nemohou zůstat neměnné např. výrobky v obchodech mohou získat nové vlastnosti nebo mohou začít patřit do nově vytvořené kategorie zboží. Tyto změny často zaváděné správcem ontologie přímo ovlivňují funkci aplikací, které tuto ontologii využívají. Proto je důležité vědět jak byly změny zdůvodněny, kdo je provedl i kdy byly provedeny a další údaje souhrnně zvané *provenance ontologie* zaznamenávat.

Tento článek se zabývá způsobem jak změny v ontologii zaznamenat a jakým způsobem je lze dotazovat. Konkrétněji se zabývá myšlenkou jak vztáhnout změny v ontologii k nějakému symbolu z ontologie.

1.1 Provenance ontologie

Slovo *provenance* pochází z francouzského slova provenir, které znamená pocházet. Toto slovo se používá k označení zdroje, původu, nebo i původce nějakého objektu. Postupem času se, ale toto slovo začalo používat i pro informace o historii objektu a obecně pro všechny události, které objekt v čase provázely. Hlavními oblastmi, kde se sledování provenance používá, jsou právní teorie (pro zabránění manipulace s důkazy), umění (ověření pravosti díla), archiválie (ověření, kdo měl k dokumentům přístup a jaké jsou jejich zdroje) i věda (citace, odkazy na prvotní původ myšlenky).

Provenance se používá i pro data. Například v datových skladech se vyznačuje, kdy a z jakého zdroje byla konkrétní data pořízena. V rozšířeném významu pak provenance pro data může označovat informace o zdrojích dat, osobách jež měly k datům přístup, aplikované transformace a algoritmy na datech, atd.. Data provenance tedy např. umožňuje přenášet důvěru ve zdroje dat a algoritmy na výstupní data.

Jak je již výše zmíněno i ontologie podléhají občas změnám, a proto je přirozené uchovávat provenanci i pro ně. Uchovávání informací k ontologii nám umožňuje:

- Zkoumat historii změn a poučit se z ní. Některé změny totiž mohou mít i nechtěný dopad a může být důležité vědět, jak byly zdůvodněny.
- Zjistit dopad konkrétních změn. Protože změna jediného axiomu v ontologii může změnit význam mnoha dalších axiomů.
- Zkontrolovat, zda provedené změny jsou oprávněné.

Samotná ontologie se skládá ze dvou částí. První část je množina tvrzení $TBOX$, která definuje tvrzení o konceptech a relacích. Příkladem tvrzení v této množině může být věta, že všichni lidé jsou smrtelní.

$$MAN \sqsubseteq MORTAL$$

Druhou částí ontologie je množina tvrzení $ABOX$, která přiřazuje konkrétní prvky ontologie ke konceptům a relacím z $TBOX$. Příkladem může být například věta, že Aristoteles je člověk.

$$Aristoteles \in MAN$$

Tvrzení z obou množin budu dále v článku nazývat axiomy ontologie.

Jelikož celá ontologie je v podstatě jen množina axiomů, tak lze změny v ontologii zúžit na změny v axiomech. Při změně v ontologii může být jeden nebo více axiomů smazáno, přidáno nebo přepsáno. Komplexnější změny ontologie mohou být složeny ze dvou základních operací – smazání axiomu a vložení nového axiomu. Změnu axiomu lze totiž nahradit odebráním jeho staré verze a vložením nové. Provenanci k ontologii lze tedy vztáhnout ke konkrétním axiomům. Jednotku provenance, která popisuje nějakou konkrétní událost nazveme provenanční atom. Jelikož s axiomy lze provést jen dvě operace, tak nám stačí k jednomu axiomu navázat dva provenanční atomy (vyznačující událost přidání axiomu a událost odebrání axiomu).

1.2 OWL anotace

Pro ukládání provenance k axiomům můžeme použít libovolnou databázi. Protože jsme v prostředí sémantického webu, tak se přímo nabízí využít jeho existující technologie a provenanci ukládat do oddělené ontologie (tedy do RDF souborů popsaných specifickou ontologií). Ukládání provenance k ontologii do oddělené ontologie s vlastním datovým modelem je výhodné, neboť umožňuje použít pro vyhledávání a odvozování v těchto informacích logického mechanismu ontologie.

Propojení provenance s původní ontologií lze realizovat dvěma způsoby. Prvním z nich je reifikace (reification) axiomů, která v nové ontologii vyjádří syntaktickou podobu původního axiomu a k této popsané struktuře pak přidává nové vlastnosti.

Z axiomu $MAN \sqsubseteq MORTAL$ se tedy stane sada axiomů

$$\begin{aligned}
&CLASS(mortal)\\
&CLASS(man)\\
&SUBCLASS\_OF\_AXIOM(axiom1)\\
&SUBCLASS\_OF\_SUPERCLASS(axiom1, mortal)\\
&SUBCLASS\_OF\_SUBCLASS(axiom1, man)\\
&SUBCLASS\_OF(man, mortal)
\end{aligned}$$

Ke kterým pak lze navazovat jednoduše vlastnosti

$$PROPERTY\_NAME(axiom1, value)$$

Jak lze snadno nahlédnout, tak tento přístup z jednoho axiomu vytvoří mnoho dalších axiomů a vede tak k výraznému zvětšení dat o provenanci a tedy i k výraznému zpomalení prohledávání a uvozování o provenanci.

Druhým přístupem je využít části zatím ještě neschváleného standardu OWL 2.0. V návrhu standardu je totiž zmíněna možnost přiřadit axiomům v originální ontologii jednoznačné URI[1]. Tím se zbavíme nutnosti popsat syntaktickou strukturu axiomu a můžeme provenanci navázat přímo na URI axiomu v originální ontologii. Oba přístupy pro ukládání provenance k ontologii jsou popsány v [7].

Samotným formátem provenance pro ontologii se článek zabývat nebude. Jaké provenanční informace se mají ukládat je totiž velmi závislé na jejich plánovaném použití a rozsah ukládaných informací se může velmi měnit. Jeden rozsáhlý datový model provenance je popsán v [5].

2 Provenance vztažená k signatuře

Pokud uchováváme provenanci k ontologii, je poměrně jednoduché se zeptat, jaké události ovlivnili konkrétní axiom z ontologie. Například je možné se zeptat, kdo konkrétní axiom zapsal a kdy se tak stalo. Tato informace může být v určitém kontextu důležitá a její nalezení je pouze dotazem na to, jaké provenanční atomy jsou svázané s URI axiomu.

V tomto článku se však snažíme o vyhodnocení podstatně složitější otázky a to, jaké události měnily význam konkrétního symbolu z ontologie. Příkladem takového dotazu může být otázka na základě jakých změn v právní ontologii ČR byl měněn význam konceptu OSVČ - osoba samostatně výdělečně činná.

Obecně lze tento dotaz formulovat, jak získat z provenance k ontologii provenanci k signatuře. *Signaturou* se uvažuje množina symbolů konceptů, relací a individuí. Dále budeme uvažovat, že provenanci vztahujeme k axiomům v ontologii a dané řešení tedy získáme tak, že najdeme co nejmenší počet axiomů, které plně určují význam symbolu (nebo signatury).

Jinou otázkou se zabývá práce [6], kde se řeší jak navázat provenanci k axiomu, který je z dané ontologie logicky odvoditelný. To se může hodit v případě, že chceme axiom z ontologie nahradit jiným, bez změny významu ontologie.

Při řešení položené otázky narazíme na dva problémy. První je skutečnost, že význam symbolu může být závislý na axiomech, které se v současné ontologii už nevyskytují. Proto je potřeba uchovávat všechny historicky používané verze ontologie. Druhým problémem je způsoben tím, že význam symbolu může být určen více axiomy a to i takovými, kde se daný symbol nevyskytuje. Z tohoto důvodu je potřeba vzít v úvahu logickou sémantiku ontologie.

Ve snaze přesněji definovat tvrzení „axiom nemění význam symbolu“ nám pomůže definice původem z oboru modularizace ontologií [4].

Definice 1 (Model konzervativní rozšíření)**.** *Nechť O a $O_1 \subseteq O$ jsou dvě $\mathcal{L}$-ontologie a* $\mathbf{S}$ *je signatura nad $\mathcal{L}$. Řekneme, že O je model* $\mathbf{S}$*-konzervativní rozšíření O_1, právě když pro každý model $\mathcal{I}$ ontologie O_1, existuje model $\mathcal{J}$ ontologie O takový, že se shoduje na interpretaci symbolů z* $\mathbf{S}$ *- formálněji $\mathcal{I}|_{\mathbf{S}} = \mathcal{J}|_{\mathbf{S}}$.*

Definice říká, že ontologie jde rozšířit přidáním nových axiomů na jinou ontologii, ale nevede to k jiné interpretaci symbolů. Lze tedy říci, že nově přidané axiomy nenesou k daným symbolům nové informace a tedy nepřispívají k jejich významu.

[1] Uniform Resource Identifier - jednotný identifikátor zdroje/objektu

Pro danou ontologii O a signaturu S lze tedy najít minimální pod-ontologie, která ještě splňují model S-konzervativní rozšíření, a říci, že axiomy v těchto ontologiích definují význam symbolů z S. Provenance pro signaturu by se tedy získala jako sjednocení provenančních atomů pro všechny tyto axiomy.

Bohužel ověření, zda ontologie je model S-konzervativní rozšíření jiné, je velmi náročné. Dokonce i pro velmi jednoduché ontologie typu $\mathcal{ALC}$ není rekurzivně spočetné. Využijeme tedy vlastnost lokality z [4] a tvrzení z [3], abychom předcházející vlastnost alespoň přibližně odhadli. Lokalitu lze vyjádřit jako syntaktické omezení na to jaké ontologie vůbec budeme posuzovat. Protože je definice lokality poměrně rozsáhlá, není zde uvedena. Zjištění vlastnosti lokality je také poměrně obtížné, ale přesto nepoměrně snazší. Například zjištění lokality pro poměrně složité ontologie definované pomocí standardu OWL DL je NEXPTIME-úplné.

Následující věta dává do vztahu vlastnost lokality a model S-konzervativního rozšíření. $Sig(O)$ označuje množinu všech symbolů konceptů, relací a individuí, které jsou použity v axiomech ontologie O.

Věta 1. *Nechť O_1, O_2 jsou dvě ontologie a $\mathbf{S}$ je signatura, takové že O_2 je lokální vůči $\mathbf{S} \cup Sig(O_1)$. Pak $O_1 \cup O_2$ je $\mathbf{S}$-model konzervativní rozšíření O_1.*

Tato věta tedy říká, že pro danou ontologii O a signaturu S stačí najít pod-ontologii O_2 ontologie O, která je lokální vůči S. Protože $O_1 \cup O_2 = O$ platí $O \setminus O_2 \subseteq O_1$ a $O \setminus O_2$ je spodním odhadem ontologie, jež lze model konzervativně rozšířit na O.

Z tohoto důvodu nezískáme veškeré provenanční informace, které mohou ovlivnit význam symbolu. Je též třeba prozkoumat, jak je daný odhad přesný a v jakém vztahu je ontologie získaná tímto způsobem k minimálním ontologiím, jež lze model konzervativně rozšířit na O.

Prezentovaný postup založený na model S-konzervativním rozšíření nebere v úvahu, že v průběhu života ontologie se používají různé verze ontologie a provenance je navázána ke všem těmto verzím. Prvně definujeme následující termín.

Definice 2 (Historie verzí ontologie)**.** *Nechť $O_1, O_2 \ldots O_N$ je posloupnost ontologií všech používaných verzí ontologie O, setříděné podle času vytvoření verze, kde O_1 je první verze ontologie a O_N poslední verze s tím, že žádná verze ontologie mezi O_i and O_{i+1} pro $i \in \{1, 2, \ldots, N-1\}$ nebyla vynechána. $O_1, O_2 \ldots O_N$ nazveme historii verzí ontologie O.*

Vyžadujeme, aby žádná používaná verze nechyběla, protože všechny provenanční atomy vztažené ke změnám v dané verzi by ve výsledku následujícího algoritmu chyběly.

Algoritmus 1. *Nechť S je signatura nad jazykem $\mathcal{L}$, $(O_i)_{i\in\{1\ldots N\}}$ historie verzí ontologie O nad $\mathcal{L}$ a $Prov(axiom)$ je funkce, která mapuje axiomy O na množinu jim příslušných provenančních atomů.*

Pro každou verzi ontologie O_i se nalezne (nebo pomocí lokality odhadne) sjednocení axiomů z množiny ontologií O‘ takových, že O‘ je model S-konzervativních rozšíření O_i. Toto sjednocení označíme E_i.

Jako E označíme sjednocení takových množin.

$$E = \bigcup_{i\in\{1,\ldots,N\}} E_i$$

Nakonec, provenanční atomy příslušející k dané signatuře S a historii verzí ontologie O získáme jako sjednocení provenančních atomů pro všechny axiomy v E.

$$Prov(S) = \bigcup_{\alpha \in E} Prov(\alpha)$$

3 Shrnutí

Článek prezentuje provenanci k ontologii, postup jak ji ukládat a algoritmus, který umožňuje vztáhnout ji k symbolům z ontologie. Protože výpočet provenance k signatuře je z definice mnohdy nemožný, je v článku navržena možnost odhadu výsledku pomocí vlastnosti zvané lokalita.

Dále bych se chtěl zaobírat, postupem jak získat pomocí lokality co nejpřesnější odhad a jak se bude na reálných datech lišit od výsledku dle definice. Také bych se chtěl zabývat možnostmi optimalizace algoritmu.

Literatura

[1] G. Antoniou and F. van Hamerlen. *A Semantic Web Primer*. The MIT Press, (2004).

[2] F. Baader, D. Calvanese, D. L. McGuiness, D. Nardi, and P. F. Patel-Schneider. *The Description Logic Handbook*. Cambridge, (2007).

[3] B. C. Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Just the right amount: Extracting modules from ontologies. In 'Sixteenth International World Wide Web Conference (WWW2007)', (2007).

[4] B. C. Grau, I. Horrocks, Y. Kazakov, and U. Sattler. *School of computer science: Modular reuse of ontologies: Theory and practice.* Journal of Artifical Inteligence Research (2008).

[5] S. Ram. The active conceptual modelling of learning workshop. In 'Space and Naval Warfare Systems Center, San Diego, May 9-12', (2006).

[6] M. Vacura and V. Svátek. Pattern-based representation and propagation of provenance metadata in ontologies. In 'EKAW 2008 Poster and Demo Proceedings', 66–68, (2008).

[7] D. Vrandečić, J. Völker, P. Haase, T. T. Duc, and P. Cimiano. A metamodel for annotations of ontology elements in owl dl. In 'Proccedings of the 2nd Workshop on Ontologies and Meta-Modeling. GI Gesellschaft für Informatik, Karlsruhe, Germany', (10 2006).

Detector simulation with Geant4[*]

Vladimír Jarý

1st year of PGS, email: jaryvlad@kmlinux.fjfi.cvut.cz
Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, CTU in Prague
advisor: Miroslav Virius, Department of Software Engineering in Economics, Faculty of Nuclear Sciences and Physical Engineering, CTU

Abstract. Monte Carlo simulations are today indispensable part of any experiment in the field of the high energy physics. This paper describes one of the most complex detector simulation toolkit called Geant4. The paper covers the most important parts of this toolkit ranging from material definition and geometry setup to primary particle generation. On the example of the transport of a photon, Monte Carlo simulation is explained.

Keywords: Geant4, simulation, Monte Carlo, C++

Abstrakt. Simulace založené na metodě Monte Carlo dnes tvoří nedílnou součást každého experimentu na poli fyziky vysokých energií. Tento článek se zabývá rozsáhlou simulační knihovnou Geant4. Článek popisuje nejdůležitější části této knihovny od definice materiálu a geometrie detektoru až po generování primárních částic. Na příkladu přenosu fotonu je vysvětlena podstata simulace metodou Monte Carlo.

Klíčová slova: Geant4, simulace, Monte Carlo, C++

1 Introduction

Simulations play an important role during the life cycle of the experiment in the particle physics. Simulations are used to design and fine–tune the detectors, to develop and test software for data analysis and acquisition, or to calculate doses of radiation.

In the first part of this article, the simulation of transport of a particle is described. The second part contains the general overview of the Geant4 simulation toolkit. Finally, the last part describes the work with this toolkit.

2 Geant4 overview

Geant4 (*GEometry ANd Tracking, version 4*) is a software toolkit used for the detector simulation. Original version of the toolkit was developed in the FORTRAN language, but in 1993 the researchers in CERN and KEK[1] independently proposed the idea to rewrite the toolkit using modern programming techniques such as the object oriented programming. These teams joined their effort in 1994 and created the RD44 project. After four years of research and development, the first Geant version based on the C++

[*] This work has been supported by the grant MSMT LA08015

[1] High Energy Accelerator Research Organization

language appeared in December 1998. More information about the history of Geant can be found in [6].

Geant4 is a free software, it can be used under the terms of the custom license [8]. The toolkit is distributed in a source–archive. Currently, Geant4 is officially supported on the three platforms: on GNU/Linux with the g++ compiler, on Mac OS X with g++, and on MS Windows XP with MS Visual Studio C++.

Today, Geant4 is used as a simulation tool in many fields including high energy physics, astrophysics, or medicine. Geant is used at all major experiments at the LHC collider at CERN, at experiment BaBar at SLAC, or in X–ray Multi–Mirror Mission at ESA. More application are showcased on the Geant4 website [12].

2.1 Monte Carlo methods

Simulations in Geant4 use the Monte Carlo methods. These methods are based on the random sampling of a variable with given probability distribution. Monte Carlo methods were developed during the Manhattan project by Stanislaw Ulam, John von Neumann, and Nicholas Metropolis. The principle of the method will be demonstrated on the transport of a particle.

Let us suppose that the particle, e. g. photon, is travelling through the infinite homogeneous medium. The mean free path of the particle is a random variable with exponential distribution. This result is known as the attenuation law [5]. Thus the probability density function of the free path can be written in the following form:

$$f(x) = \mu \cdot e^{-\mu \cdot x} \tag{1}$$

where the constant μ denotes the interaction coefficient dependant on the material. The cumulative distribution function $F(x) = \int_{-\infty}^{x} f(t)\,\mathrm{d}t$ is random variable uniformly distributed on the interval $\langle 0, 1\rangle$ ($\sim U(0,1)$) [10]. Thus, to sample the free path, it is sufficient to generate $\gamma \sim U(0,1)$ and invert the cumulative distribution function, i.e. $x_s = F^{-1}(\gamma)$. For the exponential distribution we get

$$F(x) = 1 - \exp(-\mu \cdot x) \tag{2}$$

and

$$x_s = F^{-1}(\gamma) = -\frac{\ln(1-\gamma)}{\mu} = -\frac{\ln(\bar{\gamma})}{\mu} \tag{3}$$

$\bar{\gamma}$ is also sample of uniformly distributed random variable on the interval $\langle 0, 1\rangle$ because this distribution is symmetrical about 1/2.

Typical detector consists of several layers made of different materials. In this case, the photon can pass through several layers before interaction. According to [11], let us define the number of mean free path (or number of an interaction length):

$$n_\mu = x_1 \cdot \mu_1 + x_2 \cdot \mu_2 + \cdots + x_l \cdot \mu_l \tag{4}$$

Here x_i and μ_i correspond to the step size and interaction constant in $i - th$ layer. The dimensionless number n_μ is independent of materials. At the beginning of the simulation, the value of n_μ is initialized to $-\ln(\gamma)$ ($\gamma \sim U(0,1)$). After each step, the amount of interaction length spent in the step is subtracted from n_μ. When n_μ reaches zero, interaction occurs.

In the next step of the simulation, the type of interaction is chosen. Depending on its kinetic energy, the photon can undergo one of the following processes: the photoelectric effect, the Compton scattering, and the pair production. During the photoelectric effect, the photon is absorbed by the atomic electron which is then emitted. Photoeffect is dominant at lower energies. At higher energies, the Compton scattering becomes dominant. In this process, the photon loses part of its energy and is deflected from its original direction. Photons with energy higher than $1,022\,\mathrm{MeV}$ ($1\,\mathrm{eV} = 1,602 \cdot 10^{-19}\,\mathrm{J}$) can participate in the electron–positron pair production. The ratios of these processes are known for given material and given energy; simulation program generates another random number $\gamma \sim U(0,1)$ and according to its value selects corresponding process. The simulation would continue by calculating parameters (kinetic energy, momentum, ...) of secondary particles (i. e. products of interactions) and tracking them. Again, the physical parameters are calculated by sampling some random variable.

2.2 Geant4 kernel

Geant4 covers all aspects of the detector simulation. This include geometry and material setup of the apparatus, definition of participating particles and processes, data analysis, and visualisation.

Applications written Geant4 can be described by a finite state machine with seven states. The application starts in the *PreInit* state in which it initialize itself. In the following state – *Init* – user initialization is executed. This process is discussed more deeply in the following section; when it is finished, the application switches to the *Idle* state. In this state, the application waits to the *Beam on* command which starts the simulation. The simulation is represented by two states: the *GeomClosed* and the *EventProc*. In the former state, the geometry setup and physics processes involved are locked and cannot be changed. The latter corresponds to the processing of event. These two states create the *event loop* and are also known as the *run*. When the simulation ends, the application returns to the *Idle* state. New simulation may begin or the application switches to the *Quit* state and terminates normally. In case the exception occurs, the application moves to the *Abort* state.

Basic unit of simulation in Geant4 is the event. The event begins by generating primary particle tracks. These tracks are pushed into the stack. Then the track is popped from the top of the stack and its life is simulated. Any secondary particle is also stored in the stack. The processing of event finishes when the stack is emptied. Lifetime of particle can be described by four types of objects: *Track*, *Step*, *Step point*, and *Trajectory*. Track represents a snapshot of a particle and is updated by steps. Track contains information about position, momentum, kinetic energy, proper time (time in its rest frame) of the particle, and its status. Step consists of two step points - *PreStep* and *PostStep* - and delta information - step length, increment in position, energy deposited, and others. After each step, the status of track is updated. The tracking of the particle ends when it loses its kinetic energy (and there is no rest process applicable), it leaves the area of interest, or it decays. Additionally, particle track can be deleted by user. It must be mentioned that track and step are not persistent, they are deleted at the end of the event. To store information about the lifetime of particle, one has to use *Trajectory* and *TrajectoryPoint* objects. These object copy some properties of the track and of the step. Default implementation represented by `G4VTrajectory` and `G4VTrajectoryPoint` stores only few properties. By subclassing these classes, it is possible to achieve desired behaviour.

Collection of events which share the same detector setup and the same participating particles and processes is called run.

3 Working with Geant4

3.1 General remarks

Working with Geant4 requires a decent knowledge of the C++ language and at least basic concepts of the object–oriented programming, especially the inheritance and the polymorphism. Geant4 is a large collection of classes. Many of these classes are abstract and many methods contain a dummy implementation - they do nothing. Application programmer must use the inheritance and provide his or her custom implementation in derived classes. Geant4 still does not offer its namespace, instead classes are prefixed by `G4` prefix. To ensure portability across platforms (GNU/Linux, MS Windows, Mac OS), Geant4 uses its own data types such as `G4int` or `G4double`.

Geant4 introduces several features provided by the *CLHEP*[2] library. The most notable is probably the system of physical units. Each numerical value must be accompanied by corresponding units. Additionally, Geant4 includes the `G4BestUnit` class which prints the value with the most suitable unit in given category (length, energy, density, etc). See the following listings for example:

```
G4double density = 11.35*g/cm3; //Pb
G4double pressure = 1.0*atmosphere; //atmospheric pressure
G4cout << "Step length: "
        << G4BestUnit(fStep->GetStepLength(),"Length") << G4endl;
```

Note that Geant4 replaces the output streams `cout` and `cerr` from the C++ standard template library by its own streams `G4cout` and `G4cerr`. Random number generator in Geant4 is also taken from CLHEP. Several engines (James, Ranecu) and several distributions (uniform, Gauss, Poisson) are supported.

3.2 The `main` function

Geant4 does not provide its `main` function, it is the application programmer's responsibility to write one. Several tasks must be implemented in the `main` function. At first, programmer should create an instance of the `G4RunManager` class. The run manager initializes the simulation, starts the run, and manages the event loop. To complete the

[2] A Class Library for High Energy Physics, see `http://www.cern.ch/clhep`

initialization, the programmer must define the apparatus, the particles and processes involved in the simulation, and the source of primary particles.

Detector construction To construct a model of the detector, it is necessary to subclass the abstract `G4VUserDetectorConstruction` class. This class contains one pure virtual method called `Construct`. This method must be implemented in the subclass. Detector construction includes at least definition of materials and geometry. Additionally, one can define visualisation attributes, apply magnetic field, and assign sensitive detector.

Materials in Geant4 are represented by three class: `G4Element`, `G4Isotope`, and `G4Material`. New chemical element may be constructed directly by passing its name, symbol, atomic number, and molar mass to the constructor of `G4Element` class. The other method is based on mixing of isotopes. In this case, one passes name, symbol, and number of components to the constructor and then adds the isotopes by calling `AddIsotope` method. This method takes two parameters: the isotope to be added and its relative abundance. In the following example, two elements (oxygen and enriched uranium) are created:

```
G4Element* O = new G4Element("Oxygen", "O", 8., 16.00*g/mole);
G4Isotope* U235 = new G4Isotope("U235", 92, 235, 235.01*g/mole);
G4Isotope* U238 = new G4Isotope("U238", 92, 238, 238.03*g/mole);
G4Element* U = new G4Element("Enriched uranium", "U", 2);
U->AddIsotope(U235, 90.*perCent);
U->AddIsotope(U238, 10.*perCent);
```

By mixing chemical elements, it is possible to construct molecules. It is also define new materials by mixing existing materials with elements or materials with materials. Geant4 also contains table of elements and materials provided by the National Institute of Standards and Technology [4]. In order to access this table, the `G4NistManager` class must be used; for information, see [7].

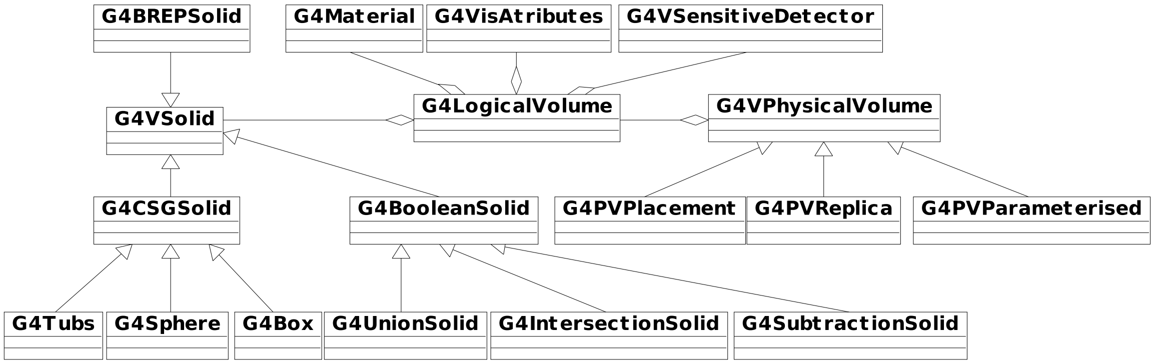


Figure 1: Simplified UML class diagram

Detector geometry consists of several volumes. Each of these volumes is described by a *solid*, a *logical volume*, and a *physical volume*. A solid defines a shape and size of a volume. Geant4 contains predefined classes for the most used shapes such as a sphere, a cuboid, or a cylinder (see Figure 1 for corresponding Geant4 classes). One can define its own solid by subclassing the `G4CSGSolid`. More complicated shapes can be created using Boolean operation: union, intersection, and subtraction. Finally, class `G4BREPSolid` provides the boundary represented shapes. Second layer, the logical volume, adds information about material, visualisation attributes, sensitive detector, magnetic field, and position of daughter physical volume. The sensitive detector is used to obtain information about passing particle(s). When the step enters the logical volume with associated sensitive detector, the `ProcessHits` method is called. The physical volume completes the definition by adding information about position and rotation of a logical volume. Logical volume can be placed once (using the `G4PVPlacement` class) or many times (`G4PVReplica` and `G4PVParameterised` classes). Volumes are organised into the mother–daughter hierarchy. Each volume, with the exception of the root volume, must have mother volume, each volume can have several daughters. Daughter volume must be fully contained within its mother volume. The root volume, also known as the *World*, defines the global coordinate system with the origin in the center. Position of a particle is given in this system. The above mentioned `Construct` method must return the pointer to the world physical volume. Because of the mother–daughter hierarchy, information about whole geometry is available through this pointer.

The geometry can be described by the GDML[3] which is a dialect of the XML [3]. GDML file stores the information about geometry in a human–readable form and allows exchanging geometry data among application; for example Java–based Graphical Geometry Editor (GGE) stores geometry in this format. Geant4 can import as well as export GDML files.

Some very complex geometries have been modeled in Geant4. For example, detector for the LHCb experiment consists of roughly 5000 logical volumes, detector for the CMS experiment of even more – 15000 volumes [2].

Particles and processes After constructing the detector, programmer must also construct list of particles, range cuts for the particle production, and physical processes involved in the simulation. The base abstract class `G4VUserPhysicsList` contains pure virtual methods `ConstructParticle`, `ConstructProcess`, and `SetCuts`. These methods must be implemented in derived subclass.

Geant4 divides particles into six categories: baryons, bosons, ions, leptons, mesons, and short-lived particles. Each particle has its own class derived from the `G4ParticleDefinition` class and each of these classes has a single object accessible via a static method. When this method is called for the first time, the corresponding single object is created. All required particles must be created in this way *before* the definition of processes. There are also six auxiliary classes that contain a method `ConstructParticle` that constructs all particles in the respective category. The following code snippet demonstrates how to create a proton and all mesons:

```
G4Proton::ProtonDefinition();
G4MesonConstructor constructor;
constructor.ConstructParticle();
```

Geant4 offers a wide range of a physical processes including transportation, electromagnetic, hadronic, and optical processes, decay, and others. Additional processes can be added by user by subclassing the `G4VProcess` class. Process can occur at any combination

[3] Geometry Description Markup Language

of the following three states: *AtRest* (e.g. positron annihilation), *AlongStep* (ionisation), and *PostStep* (decay in flight). Each particle has a process manager. It manages process in which the particle participates. New process is added by the `AddProcess` method. The following example registers three processes for an electron: multiple scattering, ionisation, and braking radiation:

```
G4String particleName = particle->GetParticleName();
if(particleName == "e-"){
  pmanager->AddProcess(new G4eMultipleScattering, -1, 1, 1);
  pmanager->AddProcess(new G4eIonisation,         -1, 2, 2);
  pmanager->AddProcess(new G4eBremsstrahlung,     -1, 3, 3);
} else if(particleName == "gamma){ ...
```

The three integer parameters in the `AddProcess` represent the ordering of the processes. The first integer corresponds to the AtRest, the second to the AlongStep, and the third to the PostStep. Value -1 means that the process is inactive in respective state.

Application programmer must also set production thresholds for certain electromagnetic processes. Bellow this threshold, no secondary track will be produced. Geant4 provides the `SetsCutsWithDefault` method that set the production threshold globally to a default value (1 mm). However, it is also possible to set different thresholds for different particles and also for different parts of the detector.

Production of primary particles In the last obligatory step of Geant4 initialization, programmer defines the source of primary tracks. This step also involves subclassing: the `G4VUserPrimaryGeneratorAction` serves as the base class. In its subclass, programmer must construct the primary generator object and implement the `GeneratePrimaries` method. This method is called at the beginning of each event. The particle generator is represented by the `G4ParticleGun` class. Via the set methods of the particle gun, it is possible to specify the number of primary particles, their type, energy, momentum, and polarisation. Particle gun does not randomize these parameters itself. The randomization, if necessary, must be implemented manually. The primary particles are generated by calling the `GeneratePrimaryVertex` of the particle gun.

Alternatively, Geant4 provides another primary generator class, the `G4HEPEvtInterface`. This generator reads and processes an ASCII file produced by some external generators (e.g. Pythia). Thus, this class works as an interface to other generators. The format of the ASCII file is described in [7].

Optional user action classes Optionally, it is also possible to set up the user interface, visualization manager, and additional user action classes. These custom user classes can be used to extract information about the run, event, or step. For example, the `G4UserRunAction` class contains the `BeginOfRunAction` method. It is called at the beginning of each run, its implementation in the `G4UserRunAction` class is empty. By overriding implementation of this method in derived class, programmer has the opportunity to modify the start of the run. For example, one can instantiate some histograms. Similarly, to modify the end of run, one has to override the `EndOfRun` method. This is a suitable place to save the histograms to a file. The other user action classes such as the G4UserEventAction or G4UserSteppingAction are described in [7].

The essential part of a typical main function can be found in the following code listing:

```
G4RunManager* runManager = new G4RunManager;
MyDetectorConstruction* detector = new MyDetectorConstruction;
runManager->SetUserInitialization(detector);
runManager->SetUserInitialization(new MyPhysicsList);
G4VUserPrimaryGeneratorAction* gen_action =
        new MyPrimaryGeneratorAction(detector);
runManager->SetUserAction(gen_action)
runManager->SetUserAction(new MyRunAction);
...
runManager->Initialize();
```

In the rest of the main function, it is possible to set up visualisation and/or user interface. Visualization may be useful for studying the geometry, detecting overlapping volumes, or publication purposes. Geant4 supports several visualisation drivers including the industry standard OpenGL. To use visualization in the application, the G4VisManager object must be constructed and initialized in the main function.

By default, the simulation process is hard–coded in the application. This means that it is necessary to modify source codes and recompile them to modify simulation parameters. This is not very flexible solution, so Geant4 offers to possibility to set up a batch mode. In this mode, the application reads and interprets a macro – a text file with instructions. Geant4 contains many commands, others can be easily implemented by user. These commands allow to start a new run, modify materials, visualisation attributes, or enable and disable certain processes. No recompilation is needed in this case. The following listings demonstrate the way in which macro file is executed:

```
G4UImanager* UI = G4UImanager::GetUIpointer();
if(argc!=1){
  G4String command = "/control/execute ";
  G4String fileName = args[1];
  UI->ApplyCommand(command+fileName);
}
```

The first command–line argument of program is taken as the location of the macro which will be executed using the command /control/execute. Moreover, Geant4 supports creating interactive applications. In such applications, users control the simulation directly from command interpreter or even from graphical user interface. Simulation thus can be modified by people who do not know the C++ language.

4 Conclusion and outlook

Geant4 is very complex toolkit for the simulation of detectors and this paper covers only the most essential aspect of Geant4. More complete description can be found in [1], [7], or [12].

The author of this paper is a member of the Joint Czech Group at the experiment COMPASS (COmmon Muon and Proton Apparatus for Structure and Spectroscopy) [9]. He is expected to use Geant4 to perform simulations of the Ring Imaging Cerenkov detector (RICH).

References

[1] S. Agostinelli et al. *Geant4–a simulation toolkit*, Nuclear Instruments and Methods in Physics Research Section A: volume 506, Issue **3**, (January 2003)

[2] M. Asai. *Geant4 tutorial course: Geometry*, Standford Linear Accelerator Center see website `http://geant4.slac.stanford.edu/SLACTutorial07/agenda.html`

[3] R. Chytracek, J. McCormick, W. Pokorski, G. Santin. *Geometry Description Markup Language for Physics Simulation and Analysis Applications*, IEEE Trans. Nucl. Sci., Vol. 53, Issue: 5, Part 2, 2892-2896, (2007)

[4] J. S. Coursey, D. J. Schwab, R. A. Dragoset. *Atomic Weights and Isotopic Composition*, National Institute of Standards and Technology, (2005)
see website `http://physics.nist.gov/PhysRefData/Compositions/index.html`

[5] I. Garbett. *Light attenuation and exponential laws*, Plus magazine, (January 2001)

[6] Geant4 collaboration. *Introduction to Geant4*, (2008)

[7] Geant4 collaboration. *Geant4 User's Guide for Application Developers*, (2009)

[8] Geant4 collaboration. *The Geant4 Software License*
see website `http://geant4.web.cern.ch/geant4/license/`

[9] G. Mallot et al. *The COMPASS experiment at CERN*, Nucl. Instrum. Methods Phys. Res., A 577 , 3 (2007) 455-518

[10] M. Virius. *Aplikace matematické statistiky – Metoda Monte Carlo*, Nakladatelství ČVUT, Praha (1998)

[11] L. Wang, S. L. Jacques, L. Zheng. *MCML – Monte Carlo modeling of light transport in multi–layered tissues*, Computer Methods and Programs in Biomedicine **47** (1995), 131–146

[12] [online], *Geant4 website*
see website `http://geant4.web.cern.ch/geant4/`

Resonant Cyclotron Acceleration of Particles by a Time Periodic Singular Flux Tube

Tomáš Kalvoda*

3rd year of PGS, email: kalvotom@fjfi.cvut.cz
Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, CTU in Prague
advisor: Pavel Šťovíček, Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, CTU

Abstract. We study the dynamics of a classical non-relativistic charged particle moving on a punctured plane under the influence of a homogeneous magnetic field and driven by a periodically time-dependent singular flux tube through the hole. We exhibit the effect of the resonance of the flux and cyclotron frequencies. The particle is accelerated to arbitrarily high energies even by a flux of small field strength. The cyclotron orbits blow up and the particle oscillates between the hole and infinity. We support this observation by an analytic study of von Zeipel first order approximation.

Abstrakt. Zabýváme se studiem klasické nerelativistické nabité částice pohybující se v propíchlé rovině pod vlivem homogenního magnetického pole a periodicky časově závislým singulárním tokem, který prochází otvorem v ploše. Odhalujeme rezonantní efekt mezi frekvencí singulárního toku a cyklotronovou frekvencí. Energie částice roste nad všechny meze i v případě že amplituda singulárního toku je malá. Během svého pohybu v rovině se částice dostane libovolně blízko k otvoru ale i libovolně daleko od něj. Toto pozorování je založeno na analýze přibližného systému získaného pomocí von Zeipelovy poruchové metody.

1 Introduction

Consider a classical point particle of mass m and charge e moving on the punctured plane $\mathbb{R}^2 \setminus \{0\}$ in the presence of a homogeneous magnetic field of magnitude b. Suppose further that a singular flux line whose strength $\Phi(t)$ is oscillating with frequency Ω pierces the origin. This is a Hamiltonian system with time-dependent Hamilton function

$$H(q,p,t) = \frac{1}{2m}\big(p - eA(q,t)\big)^2,$$

which is defined on the phase space $\mathbb{P} = \big(\mathbb{R}^2 \setminus \{0\}\big) \times \mathbb{R}^2$, and where the vector potential A is given by

$$A(q,t) = \Big(-\frac{b}{2} + \frac{\Phi(t)}{2\pi|q|^2}\Big)q^\perp,\ (q,t) \in \big(\mathbb{R}^2 \setminus \{0\}\big) \times \mathbb{R}.$$

We use a shorthand $q^\perp = (-q_2, q_1)$.

*The author wish to gratefully acknowledge support from the following grants: Grant No. LC06002 of the Ministry of Education of the Czech Republic and Grant No. 202/08/H072 of the Grant Agency of the Czech Republic.

Motivated by the rotational symmetry of the system we pass to the polar coordinates in the plane, $q = r(\cos\theta, \sin\theta)$. The new Hamiltonian in these coordinates reads

$$H(r, \theta, p_r, p_\theta, t) = \frac{1}{2m}\left(p_r^2 + \left(\frac{1}{r}\left(p_\theta - \frac{e\Phi(t)}{2\pi}\right) + \frac{eb}{2}r\right)^2\right)$$

Obviously, the angular momentum p_θ is an integral of motion. Consequently the problem reduces to the analysis of one-dimensional system. From now on we set

$$e = m = 1,$$

and thus the cyclotron frequency equals just to b. Without loss of generality it can be assumed that $b > 0$. The radial Hamiltonian, whose dynamics is to be understood, is

$$H(r, p_r, t) = \frac{1}{2}\left(p_r^2 + \left(\frac{a(t)}{r} + \frac{b}{2}r\right)^2\right), \quad a(t) = p_\theta - \frac{1}{2\pi}\Phi(t).$$

If the flux function Φ is constant, then the system is integrable. Motivated by this observation we construct a canonical transformation to the so called action-angle coordinates φ, I. The generating function of this canonical transformation is time dependent

$$\begin{aligned} S(r, I, t) = {} & \frac{1}{4}\sqrt{8bIr^2 - (br^2 - 2|a(t)|)^2} - I \arctan\left(\frac{4I - br^2 + 2|a(t)|}{\sqrt{8bIr^2 - (br^2 - 2|a(t)|)^2}}\right) \\ & - \frac{|a(t)|}{2}\arctan\left(\frac{(br^2 + 2|a(t)|)\sqrt{8bIr^2 - (br^2 - 2|a(t)|)^2}}{b^2r^4 - 4bIr^2 + 4|a(t)|^2}\right). \end{aligned}$$

The canonical transformation of variables, from (r, p_r) to anction-angle variables (φ, I), is then at each instant of time t, given by

$$r = \frac{2}{\sqrt{b}}\left(I + \frac{|a(t)|}{2} + \sqrt{I(I + |a(t)|)}\sin\varphi\right)^{1/2}, \quad p_r = \frac{2}{r}\sqrt{I(I + |a(t)|)}\cos\varphi,$$

and, conversely,

$$\varphi = -\arctan\left(\frac{1}{bp_r r}\left(p_r^2 + \frac{a(t)^2}{r^2} - \frac{b^2r^2}{4}\right)\right), \quad I = \frac{1}{2b}\left(p_r^2 + \left(\frac{|a(t)|}{r} - \frac{br}{2}\right)^2\right).$$

The new Hamiltonian reads

$$\begin{aligned} H_c(\varphi, I, t) &= H\big(r(\varphi, I, t), p_r(\varphi, I, t), t\big) + \left.\frac{\partial S(u, I, t)}{\partial t}\right|_{u = r(\varphi, I, t)} \\ &= bI - \operatorname{sign}(a(t))a'(t)\arctan\frac{\sqrt{I}\cos\varphi}{\sqrt{I + |a(t)|} + \sqrt{I}\sin\varphi}, \end{aligned}$$

and the Hamiltonian equations of motion take the form

$$\begin{aligned} \varphi &= b - \frac{\cos\varphi}{2\sqrt{I(I + |a|}}\frac{aa'}{2I + |a| + 2\sqrt{I(I + |a|}\sin\varphi}, \\ I' &= -\frac{\operatorname{sign}(a)}{2}\left(a' - \frac{|a|a'}{2I + |a| + 2\sqrt{I(I + |a|)}\sin\varphi}\right). \end{aligned}$$

Where we have suppressed the time dependence of a. We restrict ourselves to the case when $a(t)$ is a strictly positive function. More precisely, the angular momentum p_θ is supposed to be positive and much greater then amplitudes of Φ.

If we introduce

$$r_\pm = \frac{2}{\sqrt{b}}\left(I + \frac{|a(t)|}{2} \pm \sqrt{I(I+|a(t)|)}\right)^{1/2},$$

then from the coordinate transformation we see that

$$r^2 = \frac{1}{2}\big(r_+^2 + r_-^2\big) + \frac{1}{2}\big(r_+^2 - r_-^2\big)\sin\varphi.$$

Thus if φ increases then r oscillates between r_- and r_+ (though r_-, r_+ themselves are also time-dependent). Moreover, if $a(t)$ is bounded and $I \to \infty$ as $t \to \infty$ then $r_+ \to \infty$ and

$$r_-(t) = \frac{2|a(t)|}{br_+(t)} \to 0.$$

Therefore in this case the trajectory in the q-plane periodically returns to the origin and then escapes far away from it. With the growing time, on one hand, the trajectory gets closer to the origin and, on the other hand, it approaches infinity.

2 The von Zeipel averaging method

Let us first introduce necessary notation. The symbol $\mathbb{T}^d$ stands for the d-dimensional torus. For $f \in C(\mathbb{T}^d)$ and $k \in \mathbb{Z}^d$ we denote the kth Fourier coefficient of f by

$$\mathcal{F}[f]_k = \frac{1}{(2\pi)^d}\int_{\mathbb{T}^d} f(\varphi)e^{-\imath k\cdot\varphi}\mathrm{d}\varphi.$$

We introduce $\operatorname{supp}\mathcal{F}[f]$ as the set of indices corresponding to nonzero Fourier coefficients of f. For $f \in C(\mathbb{T}^d)$ and $\mathbb{L} \subset \mathbb{Z}^d$ put

$$\langle f(\varphi)\rangle_{\mathbb{L}} = \sum_{k\in\mathbb{L}} \mathcal{F}[f]_k e^{\imath k\cdot\varphi}.$$

We assume the following form of the flux function

$$\Phi(t) = 2\pi\varepsilon f(\Omega t)$$

where f is a 2π-periodic real function such that

$$\sum_{k=1}^{\infty} k\big|\mathcal{F}[f]_k\big| < \infty.$$

Hence $f \in C^1(\mathbb{T}^1)$. The coefficient $\varepsilon > 0$ is regarded as a small parameter.

We wish to study the model with the aid of the von Zeipel method which is an averaging method taking into account possible resonances (see [1] for the method, and also [5] for a general concept of the mathematical averaging theory).

2.1 Summary of basic formulas

Consider a completely integrable Hamiltonian in action-angle coordinates, $K_0(I) = \omega \cdot I$, with $I \in B \subset \mathbb{R}^d$, $\varphi \in \mathbb{T}^d$, where B is a domain in $\mathbb{R}^d$ and $\omega \in \mathbb{R}^d$ is a constant vector of frequencies. One is interested in a perturbed system with a small Hamiltonian perturbation so that the total Hamiltonian reads

$$K(\varphi, I, \varepsilon) = K_0(I) + \varepsilon K_*(\varphi, I, \varepsilon) = K_0(I) + \varepsilon K_1(\varphi, I) + \varepsilon^2 K_2(\varphi, I) + \dots .$$

The function $K_*(\varphi, I, \epsilon)$ is assumed to be analytic in all variables. The corresponding equations of motion represent a slow-fast system; the action variables vary slowly while the angle variables φ rotate with frequencies close to ω provided ε is small.

Let $\mathbb{K}$ the lattice of indices in $\mathbb{Z}^d$ corresponding to resonant frequencies and $\mathbb{K}^c$ be its complement,

$$\mathbb{K} = \{\omega\}^\perp \cap \mathbb{Z}^d, \ \mathbb{K}^c = \mathbb{Z}^d \smallsetminus \mathbb{K}.$$

In the von Zeipel method one applies a formal canonical transformation of variables $(I, \varphi) \mapsto (J, \psi)$, so that the Fourier series in the angle variables ψ of the resulting Hamiltonian $\mathcal{K}(\psi, J, \varepsilon)$ has nonzero coefficients only for indices from the lattice $\mathbb{K}$. The canonical transformation is generated by a function $S(\varphi, J, \varepsilon)$ regarded as a formal power series,

$$S(\varphi, J, \varepsilon) = \varphi \cdot J + \varepsilon S_1(\varphi, J) + \varepsilon^2 S_2(\varphi, J) + \dots,$$

and the new Hamiltonian $\mathcal{K}(\psi, J, \varepsilon)$ is also sought in the form of a formal power series,

$$\mathcal{K}(\psi, J, \varepsilon) = \mathcal{K}_0(J) + \epsilon \mathcal{K}_1(\psi, J) + \varepsilon^2 \mathcal{K}_2(\psi, J) + \dots .$$

Thus one arrives at the system of equations

$$\begin{aligned}
\mathcal{K}_0(J) &= K_0(J) = \omega \cdot J, \\
\mathcal{K}_1(\varphi, J) &= \omega \cdot \frac{\partial S_1(\varphi, J)}{\partial \varphi} + K_1(\varphi, J), \\
\mathcal{K}_j(\varphi, J) &= \omega \cdot \frac{\partial S_j(\varphi, J)}{\partial \varphi} + P_j(\varphi, J), j \geq 2,
\end{aligned}$$

where the terms P_j depend linearly on $K_1, \dots, K_j$, $\mathcal{K}_1, \dots, \mathcal{K}_{j-1}$, and on derivatives of these functions, and polynomially on $\partial S_1/\partial J, \dots, \partial S_{j-1}/\partial J$, $\partial S_1/\partial \varphi, \dots, \partial S_{j-1}/\partial \varphi$.

The formal von Zeipel Hamiltonian $\mathcal{K}(\psi, J, \varepsilon)$ is defined by the equalities

$$\begin{aligned}
\mathcal{K}_1(\psi, J) &= \langle K_1(\psi, J) \rangle_{\mathbb{K}}, \\
\mathcal{K}_j(\psi, J) &= \langle P_j(\psi, J) \rangle_{\mathbb{K}}, \ \text{for } j \geq 2.
\end{aligned}$$

Coefficients $S_j(\varphi, J)$ of the generating function $S(\varphi, J, \varepsilon)$ are then solutions of the first order differential equations

$$\begin{aligned}
\omega \cdot \frac{\partial S_1(\varphi, J)}{\partial \varphi} &= -\langle K_1(\varphi, J) \rangle_{\mathbb{K}^c}, \\
\omega \cdot \frac{\partial S_j(\varphi, J)}{\partial \varphi} &= -\langle P_j(\varphi, J) \rangle_{\mathbb{K}^c}, \ \text{for } j \geq 2.
\end{aligned}$$

In practise one truncates $\mathcal{K}(\psi, J, \varepsilon)$ at some order $m \geq 1$ of the parameter ε; this defines the mth order averaged Hamiltonian

$$\mathcal{K}_{(m)}(\psi, J, \varepsilon) = \mathcal{K}_0(J) + \varepsilon\mathcal{K}_1(\psi, J) + \ldots + \varepsilon^m\mathcal{K}_m(\psi, J).$$

Similarly, let $S_{(m)}(\varphi, J, \varepsilon)$ be the truncated generating function. If $(\psi(t), J(t))$ is a solution of the Hamiltonian equations for $\mathcal{K}_{(m)}(\psi, J, \varepsilon)$, and if $(\varphi(t), I(t))$ is the same solution after the inverted canonical transformation generated by $S_{(m)}(\varphi, J, \varepsilon)$, then $(\varphi(t), I(t))$ is expected to approximate well the solution of the original system (governed by the Hamiltonian $K(\varphi, I, \varepsilon)$ for times of order $1/\varepsilon^m$ (see [1] for a detailed discussion).

Suppose there exists a basis of the lattice $\mathbb{K}$ over $\mathbb{Z}$ formed by integer vectors $r_1, \ldots, r_s$. An important fact is that the von Zeipel Hamiltonian $\mathcal{K}_{(m)}(\psi, J, \varepsilon)$ (for any m) has additional $d-s$ integrals of motion which are linear combinations with integer coefficients of the action variables $J_1, \ldots J_d$. In fact, let $\mathbf{R}$ be a unimodular $d \times d$ matrix with integer entries such that its first s rows coincide with the vectors r_j (such a matrix is known to exist, see [2]). Consider yet another canonical transformation of coordinates, $(J, \psi) \mapsto (L, \chi)$, generated by the function $\tilde{S}(\psi, L) = L \cdot \mathbf{R}\psi$. Hence $\chi = \mathbf{R}\psi$, $J = \mathbf{R}^{\mathrm{T}}L$. The resulting Hamiltonian depends only on the first s angles $\chi_1, \ldots, \chi_s$, and so the momenta $L_{s+1}, \ldots, L_d$ are integrals of motion.

3 Dynamics of the Averaged System

In order to apply the von Zeipel method to the problem at hand we first pass to the extend phase space by introducing a new phase $\varphi_2 = \Omega t$ and its conjugate momentum I_2. The old variables φ, I are redenoted as φ_1, I_1. The Hamiltonian on the extended phase space is defined as

$$K(\varphi_1, \varphi_2, I_1, I_2) = \Omega I_2 + H_c(\varphi_1, I_1, \varphi_2/\Omega).$$

The systems of Hamiltonian equations for H_c and K are equivalent provided the initial conditions are properly matched (if $\varphi(0) = \varphi_0$ on the original phase space then $(\varphi_1(0), \varphi_2(0)) = (\varphi_0, 0)$ on the extended phase space). To adjust the notation to the general scheme, as explained above, we also set

$$\omega_1 = b, \quad \omega_2 = \Omega.$$

Thus one starts from the Hamiltonian on the extended phase space

$$K(\varphi, I, \varepsilon) = \omega_1 I_1 + \omega_2 I_2 + \varepsilon F(\varphi, I, \varepsilon)$$

where

$$F(\varphi, I, \varepsilon) = \omega_2 f'(\varphi_2) \arctan\left(\frac{\sqrt{I_1}\cos\varphi_1}{\sqrt{I_1 + p_\theta - \varepsilon f(\varphi_2)} + \sqrt{I_1}\sin\varphi_1}\right).$$

We assume that $0 < \varepsilon \ll p_\theta$.

If the ratio ω_2/ω_1 is irrational then the lattice $\mathbb{K}$ is trivial, $\mathbb{K} = \{0\}$, and the von Zeipel method reduces to the ordinary averaging method in angle variables φ. The averaged Hamiltonian depends only on action variables I and trajectories are then obviously bounded. Instead we focus on the case when

$$\nu := \frac{\omega_2}{\omega_1} = \frac{p}{q}$$

and $p, q \in \mathbb{N}$ are coprime. As we shall see, a resonance is exhibited already in the first order of the von Zeipel method to which we restrict our discussion.

We have

$$K(\varphi, I, \varepsilon) = \omega_1 I_1 + \omega_2 I_2 + \varepsilon K_1(\varphi, I) + \varepsilon^2 \tilde{K}(\varphi, I, \varepsilon)$$

where $\tilde{K}(\varphi, I, \varepsilon)$ is an analytic function in ε,

$$K_1(\varphi, I) = \omega_2 f'(\varphi_2) F_1(\varphi_1, I_1),$$

and

$$F_1(\varphi_1, I_1) = \arctan\left(\frac{\sqrt{I_1}\cos\varphi_1}{\sqrt{I_1 + p_\theta} + \sqrt{I_1}\sin\varphi_1}\right).$$

For $|\beta| < 1$,

$$\mathcal{F}[F_1(\varphi_1, I_1)]_k = \frac{\imath^{k-1}}{2k}\left(\frac{I_1}{I_1 + p_\theta}\right)^{|k|/2}, \text{ for } k \neq 0, \quad \mathcal{F}[F_1(\varphi_1, I_1)]_0 = 0.$$

Obviously, the Fourier image of $K_1(\varphi, I)$ takes nonzero values only for indices (k, l), $k \in \mathbb{Z} \smallsetminus \{0\}$, $l \in \operatorname{supp}\mathcal{F}[f] \smallsetminus \{0\}$, and

$$\mathcal{F}[K_1(\varphi, I)]_{(k,l)} = \imath l \omega_2 \mathcal{F}[f(\varphi_2)]_l \mathcal{F}[F_1(\varphi_1, I_1)]_k.$$

Next we proceed to the von Zeipel canonical transformation of the first order. Set

$$\beta = \beta(J_1) = \sqrt{\frac{J_1}{J_1 + p_\theta}}.$$

The resonant lattice is given by $\mathbb{K} = \mathbb{Z}(p, -q)$, and one has

$$\begin{aligned}\mathcal{K}_1(\psi, J) &= \sum_{m \in \mathbb{K}} \mathcal{F}[K_1(\psi, J)]_m e^{\imath m \cdot \psi} \\ &= -\frac{\omega_2}{2} \sum_{n \in \mathbb{Z} \smallsetminus \{0\}} \mathcal{F}[f]_{-nq} \imath^{np} \beta(J_1)^{p|n|} e^{\imath n(p\psi_1 - q\psi_2)}.\end{aligned}$$

$S_1(\varphi, J)$ is a solution to the differential equation

$$\omega \cdot \frac{\partial S_1(\varphi, J)}{\partial \varphi} = -K_1(\varphi, J) + \mathcal{K}_1(\varphi, J).$$

Seeking $S_1(\varphi, J)$ in the form

$$S_1(\varphi, J) = \sum_{k=1}^{\infty} \mathcal{F}[f']_k \Big(G_k(\varphi_1, J_1) + \overline{G_k(\varphi_1, J_1)} e^{-\imath k \varphi_2}\Big)$$

one finally arrives at the countable system of equations

$$\left(\frac{\partial}{\partial \varphi_1} + \imath k \nu\right) G_k(\varphi_1, J_1) = \nu \sum_{n \in \mathbb{Z} \smallsetminus \{0\}, n \neq -k\nu} \frac{\imath^{n+1}}{2n} \beta(J_1)^{|n|} e^{\imath n \varphi_1}.$$

For the solution we choose

$$G_k(\varphi_1, J_1) = \nu \sum_{n\in\mathbb{Z}\smallsetminus\{0\},n\neq -k\nu} \frac{\imath^n}{2n(n+k\nu)}\beta(J_1)^{|n|}e^{\imath n\varphi_1}.$$

Of course, if $k\nu \notin \mathbb{Z}$ then the restriction $n \neq -k\nu$ is void. On the other, if $k\nu \in \mathbb{Z}$, and this happens if and only if $k \in \mathbb{Z}q$, then the solution $G_k(\varphi_1, J_1)$ is not unique.

Thus one finds the von Zeipel Hamiltonian of the first order,

$$\mathcal{K}_{(1)}(\psi, J) = \frac{\omega_1}{q}(qJ_1 + pJ_2) + \varepsilon\mathcal{K}_1(\psi, J).$$

Since p and q are coprime there exist $r, s \in \mathbb{Z}$ such that $sp + rq = 1$. Put

$$\mathbf{R} = \begin{pmatrix} p & -q \\ r & s \end{pmatrix}$$

and consider the canonical transformation $\chi = \mathbf{R}\psi$, $J = \mathbf{R}^{\mathrm{T}}L$. In particular,

$$\chi_1 = p\psi_1 - q\psi_2,\ L_2 = qJ_1 + pJ_2,\ J_1 = pL_1 + rL_2.$$

The momentum L_2 is an integral of motion for the Hamiltonian $\mathcal{K}_{(1)}(\psi, J)$. Let us define

$$\mathcal{Z}(\chi_1, J_1) = \varepsilon p\mathcal{K}_1(\mathbf{R}^{-1}\chi, J).$$

Then

$$\chi_1'(t) = \frac{\partial\mathcal{Z}(\chi_1, J_1)}{\partial J_1},$$
$$J_1'(t) = -\frac{\partial\mathcal{Z}(\chi_1, J_1)}{\chi_1}$$

Thus the evolution in coordinates χ_1, J_1 is governed by the Hamiltonian $\mathcal{Z}(\chi_1, J_1)$.

Set

$$h(z) = -\varepsilon p\omega_1 \sum_{n=1}^{\infty} \mathcal{F}[f]_{-nq}\imath^{np}z^n$$

and

$$\rho(x) = \beta(x)^p = \left(\frac{x}{x+p_\theta}\right)^{p/2}, \quad x > 0.$$

Then $h(z)$ is holomorphic on the open unit disk $B_1 \subset \mathbb{C}$ and according to our assumptions $h \in C^1(\overline{B_1})$. One has $\mathcal{Z}(\chi_1, J_1) = \mathrm{Re}[h(\rho(J_1)e^{\imath\chi_1})]$. We will investigate the dynamics generated by such a Hamiltonian in a more general setting.

3.1 General results

In this subsection we assume the following. Let $\rho :]0, +\infty[\to]0, 1[$ be continuously differentiable function such that $\rho'(x) > 0$ for all $x > 0$ and $\lim_{x\to+\infty} \rho(x) = 1$ and $\rho(0_+) = 0$.

Our aim is to investigate the dynamics of a Hamiltonian system with Hamilton function defined on $\mathbb{R}\times]0,+\infty[$ by

$$\mathcal{Z}(\chi, J) = \mathrm{Re}\big[h\big(\rho(J)e^{\imath\chi}\big)\big], \tag{1}$$

where h is a nonconstant holomorphic function on B_1 and $h \in C^1(\overline{B_1})$. The corresponding Hamiltonian equations of motion can be written in the following form

$$\begin{aligned} \dot{\chi} &= \frac{\partial\mathcal{Z}}{\partial J} = \frac{\rho'(J)}{\rho(J)}\mathrm{Re}\big[zh'(z)\big], \\ \dot{J} &= -\frac{\partial\mathcal{Z}}{\partial\chi} = \mathrm{Im}\big[zh'(z)\big]r, \end{aligned} \tag{2}$$

where $z = \rho(J)e^{\imath\chi}$.

Lemma 1. *Let $\Omega \subset \mathbb{C}$ be a domain and f holomorphic in Ω. If γ is a closed path in Ω and* Ref *is constant along γ then f is constant in Ω*

Proof. Since Ref is harmonic in the interior of γ, denoted int γ, Ref is constant also in the int γ. According to the Cauchy-Riemann equations Imf is constant in int γ. Hence f is constant in int γ and consequently in Ω. □

Theorem 1. *Let h and ρ be as above and $\mathcal{Z}(\chi, J)$ as in* (1)*. Then for almost all initial data $(\chi(0), J(0))$ the corresponding Hamiltonian trajectory fulfils*

$$\lim_{t\to+\infty}\chi(t) = \chi(\infty) \in \mathbb{R}, \quad \lim_{t\to+\infty} J(t) = +\infty, \tag{3}$$

and

$$\lim_{t\to+\infty} J(t) = \mathrm{Im}\big[e^{\imath\chi(\infty)}h'\big(e^{\imath\chi(\infty)}\big)\big] > 0. \tag{4}$$

Proof. Set $R(z) = \mathrm{Re}[h(z)]$ for $z \in \overline{B_1}$. Then $D_zR \equiv \big(\mathrm{Re}[h'[z]], -\mathrm{Im}[h'[z]]\big)$. Hence $D_zR = 0$ if and only if $h'(z) = 0$, and the set of critical points of R in B_1 is at most countable and has no accumulation points in B_1. By the Sard theorem, almost all $y \in \mathbb{R}$ are regular values of $R \restriction \partial B_1$. If y is a regular value both of R and $R \restriction \partial B_1$ then the level set $R^{-1}(y)$ is a compact one-dimensional C^1 submanifold with boundary of $\overline{B_1}$. Moreover,

$$\partial\big(R^{-1}(y)\big) = R^{-1}(y) \cap \partial B_1,$$

$R^{-1}(y)$ is not tangent to ∂B_1 at any point, and $R^{-1}(y) \cap B_1$ is a smooth submanifold of B_1 (for more details see [4, 3]). By the classification of compact connected one-dimensional manifolds [3], every component of $R^{-1}(y)$ is diffeomorphic either to a circle or to a closed interval. But the first possibility is excluded by the Lemma 1. Thus every component Γ of $R^{-1}(y)$ is diffeomorphic to a closed interval, $\partial\Gamma = \{a, b\} = \Gamma \cap \partial B_1$, and Γ is not tangent to ∂B_1 neither at a nor at b.

Let $z \in B_1$ be such that $D_zR \neq 0$. By the local submersion theorem [3], R is locally equivalent at z to the canonical submersion

$$\mathbb{R}^2 \ni (x, y) \mapsto x \in \mathbb{R}.$$

Hence z possesses an open neighborhood U such that $R(U)$ is an open interval and almost every $y \in R(U)$ is a regular value both of R and $R \upharpoonright \partial B_1$. By the Fubini theorem[1], for almost every $w \in U$, $R(w)$ is a regular value both of R and $R \upharpoonright \partial B_1$. The same claim is true for almost all $w \in B_1$ because the set of critical points of R in B_1 is at most countable. It follows that for almost all $(\chi, J) \in \mathbb{R}\times]0,+\infty[$, $R(\rho(J)e^{\imath\chi})$ is a regular value both of R and $R \upharpoonright \partial B_1$.

Suppose now that an initial condition $(\chi(0), J(0))$ has been chosen so that $y = R\big(\rho(J(0))e^{\imath\chi(0)}\big)$ is a regular value both of R and $R \upharpoonright \partial B_1$. Let Γ be the component of $R^{-1}(y)$ containing the point $\rho\big(J(0)\big)e^{\imath\chi(0)}$. Since the Hamiltonian $\mathcal{Z}(\chi, J)$ is an integral of motion the Hamiltonian trajectory $z(t) = \rho(J(t))e^{\imath\chi(t)}$ is constrained to the submanifold $\Gamma \subset \overline{B_1}$. We have to show that $z(t)$ reaches the boundary ∂B_1 as $t \to +\infty$. The tangent vector to the trajectory at the point $z(t)$ equals

$$\frac{\mathrm{d}z(t)}{\mathrm{d}t} = \imath\rho\big(J(t)\big)\rho'\big(J(t)\big)\overline{h'\big(z(t)\big)}.$$

Since $\rho'(J) > 0$ for all $J > 0$ and $h'(z)$ has no zeros on Γ (because y is a regular value) it follows that $z(t)$ leaves any compact subset of B_1 in a finite time. It remains to show that $z(t)$ does not reach ∂B_1 in finite time. But by equation of motion (2)

$$|J'(t)| \leq \max_{z\in\partial B_1} |h'(z)|$$

and so $J(t)$ cannot grow faster than linearly.

This reasoning clearly shows that (3) is valid. Using the equations of motion once more one can deduce (4). Obviously the limit must be nonnegative. Denote $\partial R = R \upharpoonright B_1$. Then ∂R can be regarded as a function of the angle variable, $\partial R(x) = \mathrm{Re}[h(e^{\imath x})]$, and one has

$$\partial R'(\chi(\infty)) = -\mathrm{Im}\big[e^{\imath\chi(\infty)}h'\big(e^{\imath\chi(\infty)}\big)\big] \neq 0$$

because $y = \partial R(\chi(\infty))$ is a regular value of ∂R. □

3.2 Conclusion

It is now straightforward to apply the results of the preceding subsection to our problem. Remember first that one has to apply the inverted canonical transformation, from (ψ, J) to (φ, I),

$$\psi = \varphi + \varepsilon\frac{\partial S_1(\varphi, J)}{\partial J}, \;\; I = J + \varepsilon\frac{\partial S_1(\varphi, J)}{\partial \varphi}.$$

In the present case, $\omega_2/\omega_1 = p/q$, p and q are coprime and q is such that

$$\mathrm{supp}\,\mathcal{F}[f] \cup \big(\mathbb{Z}q \smallsetminus \{0\}\big) \neq \emptyset,$$

so h is nonzero and the averaged system is nontrivial, and we can use the results of Theorem 1. By a tedious calculation it is possible to estimate the partial derivatives of

[1] Let $n = k + l$, A be a closed subset of $\mathbb{R}^n = \mathbb{R}^k \times \mathbb{R}^l$, and P_l the canonical submersion of $\mathbb{R}^n$ into $\mathbb{R}^l$. If $P_l(A \cap (\{c\} \times \mathbb{R}^l))$ has a measure zero in $\mathbb{R}^l$ for each $c \in \mathbb{R}^k$ then A has measure zero in $\mathbb{R}^n$.

S_1 and obtain the asymptotic behaviour of the original action-angle coordinates. More precisely, in the resonant case and for almost all initial conditions $(\varphi_1(0), I_1(0))$,

$$\lim_{t\to+\infty} \big(\varphi_1(t) - \omega_1 t\big) = \alpha \in \mathbb{R}, \quad \lim_{t\to+\infty} \frac{I_1(t)}{t} = C > 0.$$

Therefore, in this case the trajectory in the q-plane can be described exactly as at the end of Section 1.

References

[1] V. I. Arnold, V. V. Kozlov, and A. I. Neishtadt. *Mathematical Aspects of Classical and Celestial Mechanics, Dynamical Systems III.* Encyclopaedia Math. Sci. 3. Springer, Berlin, (1993).

[2] J. W. S. Cassels. *An Introduction to Diophantine Approximation.* Cambridge Tracts in Mathematics and Mathematical Physics, no. 35. Cambridge University Press, New York, (1957).

[3] V. Guillemin and A. Pollack. *Differential topology.* Prentice-Hall, Engleswood Cliffs, New Jersey, (1974).

[4] M. W. Hirsch. *Differential Topology.* Springer, New York, (1994).

[5] J. A. Sanders and F. Verhulst. *Averaging Methods in Nonlinear Dynamical Systems.* Springer, New York, (1985).

Non-standard representations of p-adic numbers*

Karel Klouda

3rd year of PGS, email: karel@kloudak.eu
Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, CTU in Prague
advisor: Edita Pelantová, Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, CTU in Prague
Christiane Frougny, LIAFA, CNRS and Université Paris 8

Abstract. We study a non-standard numeration system for p-adic numbers which is based on the rational base system proposed by S. Akiyama, C. Frougny, and J. Sakarovitch. We also briefly introduce p-adic version of β-expansions.

Abstrakt. Zkoumáme nestandardní numerační systém pro p-adická čísla, který je založen na systému navrženém S. Akiyamou, C. Frougny, and J. Sakarovitchem. Dále také zmíníme p-adickou obdobu β-rozvojů.

1 Introduction

The field of p-adic numbers, denoted by $\mathbb{Q}_p$, is an extension of the field $\mathbb{Q}$ of rational numbers in a way complementary to the classical extension: the field $\mathbb{R}$ of real numbers. The letter p refers to a prime number p and so there exist an infinite number of p-adic fields, each corresponding to one prime number. The topological structure of p-adic fields and of the field of real numbers are very different; in fact, one can find the topology of $\mathbb{Q}_p$ very nonintuitive. Nevertheless, it is still reasonable to be concerned with such an unusual construction since, due to the celebrated Ostrowski's theorem from 1918, the p-adic fields and the field of real numbers are in some sense all possible completions of $\mathbb{Q}$. Although the Ostrowski's theorem clarified the full significance of p-adic numbers, they have been introduced and systematically studied earlier by Hensel; his first work [4] on this topic is from the year 1897.

Since the construction of $\mathbb{Q}_p$ is analogous to $\mathbb{R}$, there exists a good reason to study number theory at all these completions simultaneously. The theory of p-adic numbers serves as a useful tool in solving many problems of number theory. Many objects from real analysis has its p-adic analogies. Furthermore, imitating the construction of $\mathbb{C}$ from $\mathbb{R}$, one can construct $\mathbb{C}_p$, the p-adic analogue of the complex numbers.

There is a rich literature devoted to the p-adic analysis and number theory. Nice historical review and further references can be found in [5], very friendly and accessible introduction to p-adic numbers is [3]. Our aim is to propose new non-standard way how to represent p-adic numbers; the standard way is a representation in base p with digits in the alphabet $\mathcal{A}_p = \{0, 1, \ldots, p-1\}$. Every p-adic number has then unique such representation in the form of a left infinite word over $\mathcal{A}_p$. We will study a possibility of using rational base

*This work has been suported by the Czech Science Foundation grant 201/09/0584 and by the grant LC06002 of the Ministry of Education, Youth, and Sports of the Czech Republic.

representation. Our starting point will be the rational base number system introduced in [1]. This system turns out to be natural generalization of the standard one. We will find answers to questions usually connected with numeration systems:

1. How many representations of a given number exist?

2. Which numbers have finite representation?

3. Which numbers have (eventually or purely) periodic representation?

1.1 Construction of the field of p-adic numbers $\mathbb{Q}_p$

As we said above, the construction of $\mathbb{Q}_p$ is analogous to the construction of $\mathbb{R}$; as well as $\mathbb{R}$ is a completion of $\mathbb{Q}$ with respect to the classical absolute value, the set $\mathbb{Q}_p$, p prime, is defined as a completion of $\mathbb{Q}$ with respect to the *p-adic absolute value.*

Definition 1. *Let p be a prime number. The* p-adic valuation *on $\mathbb{Z}$ is the function $v_p : \mathbb{Z} \setminus \{0\} \to \mathbb{R}$ given by*

$$n = p^{v_p(n)} n' \quad \textit{with} \quad p \nmid n'.$$

The extension to the set of rational numbers is as follows: for $x = \frac{a}{b} \in \mathbb{Q}$

$$v_p(x) = v_p(a) - v_p(b).$$

And, finally, the p-adic absolute value *on $\mathbb{Q}$ is defined by*

$$|x|_p = -p^{v_p(x)}.$$

One can say that the value $v_p(x)$ measures “divisibility” of x by p. To make this tricky notion a bit clearer let us consider several examples: $v_p(p^n) = n$ and so $|p^n| = p^{-n}$ and p^n converges to 0; if q is a prime number different from p, then $v_p(q^n) = 0$ and $|q^n| = p^{-0} = 1$; if $x = p_1^{a_1} \cdots p_k^{a_k}$, where p_i are prime factors of x, then $|x|_{p_i} = p^{-a_i}$ and $|x|_q = 0$ for all other primes q.

Recalling that two absolute values are equivalent if they define the same topology, we can say, by the following theorem, that we have found all the absolute values on $\mathbb{Q}$.

Theorem 2 (Ostrovski)**.** *Every non-trivial absolute value on $\mathbb{Q}$ is equivalent to the classical absolute value $|\ |$ or to one of the absolute values $|\ |_p$, where p is prime.*

For the proof and other details on p-adic numbers see [3].

1.2 Standard representation of p-adic numbers

Standard and well studied way how to represent p-adic numbers is the representation in the form of a power series in p.

Theorem 3. *Every $x \in \mathbb{Q}_p$ can be written in the form*

$$\begin{aligned} x &= b_{-k_0} p^{-k_0} + \cdots + a_0 + a_1 p + a_2 p^2 + \cdots + a_k p^k + \cdots \\ &= \sum_{k \geq -k_0} a_k p^k \end{aligned}$$

with $a_k \in \mathcal{A}_p$ and $-k_0 = v_p(x)$. This representation $< x >_p = \cdots a_2 a_1 a_0 \bullet a_{-1} \cdots a_{-k_0}$ of the form of a left infinite word over $\mathcal{A}_p$ is unique.

Of course, the infinite sum converges to x only with respect to the p-adic absolute value. There are several ways how to calculate the word $< x >_p$. The most convenient for our purposes is the following algorithm:

Algorithm 4. *Let*[1] $x \in \mathbb{Z} \setminus \{0\}$*, put* $s_0 := x$ *and for all* $i \in \mathbb{N}$ *define* s_{i+1} *by*

$$s_i = ps_{i+1} + a_i, \quad a_i \in \mathcal{A}_p.$$

Hence we have for all $n > 0$

$$x = s_0 = s_1p + a_0 = s_2p^2 + a_1p + p = \cdots = s_np^n + \sum_{k=1}^{n-1} a_kp^k.$$

It is easy to show that the sequence s_n is bounded, i.e. eventually periodic (for positive x it is even eventually zero), and so

$$\left| x - \sum_{k=1}^{n-1} a_kp^k \right|_p = |s_n|_p|p^n|_p = |s_n|_pp^{-n} \to 0 \quad \text{as } n \to \infty.$$

Hence, we know how to obtain the representation of integers; however, the algorithm can be easily modified for rational $x = \frac{s}{t}$, where s, t are the lowest terms:

Algorithm 5. *Let* $x = \frac{s}{t}$*,* p *and* t *mutually prime. Put* $s_0 := s$ *and for all* $i \in \mathbb{N}$ *define* s_{i+1} *by*

$$\frac{s_i}{t} = p\frac{s_{i+1}}{t} + a_i, \quad a_i \in \mathcal{A}_p.$$

If t and p are not mutually prime, i.e., $v_p(s/t) < 0$, multiply x by p until x can be written as $\frac{sp^\ell}{t'}$ with t' co-prime to p. Then apply the algorithm obtaining $< xp^\ell >_p = \cdots a_2a_2a_0$. Then, clearly, $< x >_p = \cdots a_{\ell+1}a_\ell \centerdot a_{\ell-1} \cdots a_0$. Thus there is no lost of generality.

As in the case of integral x, s_n is eventually periodic (but not eventually zero!). Moreover, employing the fact that in $\mathbb{Q}_p$ we have $\sum_{n\geq 0} p^n = \frac{1}{1-p}$, it can be proved that each eventually periodic word represents some rational number. Putting all this together, we have answers to all three questions from the introduction:

Theorem 6. *Let* $x \in \mathbb{Q}_p$*. Then* $< x >_p$ *is*

1. *uniquely given,*

2. *finite if, and only if,* $x \in \mathbb{N}$*,*

3. *eventually periodic if, and only if,* $x \in \mathbb{Q}$*.*

[1] As usual, zero is represented by the empty word ε.

2 Representation of p-adic numbers in rational base

Every real number x can be written as a power series in an integer $b > 1$ so that $x = \sum_{k \leq k_0} a_k b^k$, where $a_i \in \mathcal{A}_b = \{0, 1, \dots, b-1\}$. We say that x is represented by the right infinite word $a_{k_0} a_{k_0 - 1} \cdots a_0 a_{-1} \cdots$. Regarding answers to our three questions for this numeration system, they are all very similar for all values of b. As we will see, it is not the case for p-adic numbers.

There are several generalizations of this classical integer base system. Very famous one arises if the integer $b > 1$ is replaced by a general real number $\beta > 1$ and the alphabet by $\mathcal{A}_{\lfloor \beta \rfloor}$; the result is co-called β-expansion proposed by Rényi [6] (for details see, e.g, [2]). The possibility of introducing an analogue of β-expansion for p-adic numbers is a subject of the last section. However, our main results apply to another generalization proposed in [1], which we will now describe.

2.1 MD algorithm

In what follows we assume that $p > q > 0$ are co-prime positive integers. Let us consider the following algorithm introduced in [1] and named *modified division (MD) algorithm*[2]:

Algorithm 7. *Let s be a nonzero integer and t a positive integer co-prime to both p and q. Put $s_0 := s$ and for all $i \in \mathbb{N}$ define s_{i+1} by*

$$\frac{qs_i}{t} = \frac{ps_{i+1}}{t} + a_i, \quad a_i \in \mathcal{A}_p. \tag{1}$$

The uniquely given word $\cdots a_2 a_1 a_0$ is called $\frac{1}{q}\frac{p}{q}$-representation of $x = \frac{s}{t}$ and denoted by $< x >_{\frac{1}{q}\frac{p}{q}}$.

Example 8. *Let $p = 3, q = 2$, then:*
$< 5 >_{\frac{1}{q}\frac{p}{q}} = 2101$ *with* $(s_i)_{i \geq 0} = 5, 3, 2, 1, 0, 0, \cdots$,
$< -5 >_{\frac{1}{q}\frac{p}{q}} = \cdots 2222102$ *with* $(s_i)_{i \geq 0} = -5, -3, -2, -2, -2, \cdots$,
$< 11/4 >_{\frac{1}{q}\frac{p}{q}} = 201$ *with* $(s_i)_{i \geq 0} = 11, 6, 4, 0, 0, \cdots$,
$< 11/8 >_{\frac{1}{q}\frac{p}{q}} = \cdots 111111222$ *with* $(s_i)_{i \geq 0} = 11, 2, -4, -8, -8, -8, \cdots$,
$< 11/5 >_{\frac{1}{q}\frac{p}{q}} = \cdots 020202022112$ *with* $(s_i)_{i \geq 0} = 11, 4, 1, -1, -4, -6, -4, -6, \cdots$.

For each $n > 0$ we have

$$\frac{s}{t} = \frac{p}{q}\frac{s_1}{t} + \frac{a_0}{q} = \cdots = \left(\frac{p}{q}\right)^n \frac{s_n}{t} + \sum_{k=0}^{n-1} \frac{a_k}{q} \left(\frac{p}{q}\right)^k,$$

thus

$$\frac{s}{t} - \sum_{k=0}^{n-1} \frac{a_k}{q} \left(\frac{p}{q}\right)^k = \left(\frac{p}{q}\right)^n \frac{s_n}{t}.$$

It means that the sum converges in $\mathbb{R}$ (i.e., with respect to the classical absolute value $| \; |$) to s/t if and only if $(s_i)_{i \geq 0}$ is eventually zero. But we have learned that there are

[2] Actually, in the article the MD algorithm is defined only for integers, i.e., only for $t = 1$.

other absolute values definable on $\mathbb{Q}$, namely the p-adic absolute values. In order for the sum converges to s/t, the sequence $\left(\frac{p}{q}\right)^n$ must converge to zero (note that $(s_i)_{i\geq 0}$ is again bounded). But it happens only with respect to absolute values $|\ |_r$, where r is a prime factor of p (we did *not* assume p is prime!).

2.2 $\frac{1}{q}\frac{p}{q}$-representation of integers

The case of positive integers is well studied in [1]: $\frac{1}{q}\frac{p}{q}$-representation of a positive integer is always finite since $(s_i)_{i\geq 0}$ is eventually zero. On the other hand, if we start with negative s_0, then s_i is negative for all i and so $< s_0 >_{\frac{1}{q}\frac{p}{q}}$ is infinite (= not ending in infinite sequence of zeros). In fact, we can prove even more:

Lemma 9. *Let $s \in \mathbb{N} \setminus \{0\}$. Then:*

(i) $< s >_{\frac{1}{q}\frac{p}{q}} = a_n \cdots a_1 a_0$ is finite and

$$s = \sum_{k=0}^{n} \frac{a_k}{q} \left(\frac{p}{q}\right)^k,$$

(ii) $< -s >_{\frac{1}{q}\frac{p}{q}} = \cdots a_2 a_1 a_0$ is eventually periodic with period 1, i.e., $a_{n+k} = b$ for some n and all $k \geq 0$ and

$$-s = \sum_{k=0}^{\infty} \frac{a_k}{q} \left(\frac{p}{q}\right)^k$$

in $\mathbb{Q}_r$ if, and only if, r is a prime factor of p.

Moreover, if $s(p-q) \leq p-1$, then $b = s(p-q)$, otherwise $b = \left\lfloor \frac{p-1}{p-q} \right\rfloor (p-q)$.

2.3 Finite $\frac{1}{q}\frac{p}{q}$-representation

Let

$$x = \sum_{k=0}^{n} \frac{a_k}{q} \left(\frac{p}{q}\right)^k,$$

then, clearly, $x = \frac{m}{q^{n+1}}$ for some $m \in \mathbb{N}$. Hence, if x has a finite $\frac{1}{q}\frac{p}{q}$-representation of length n, then it is of the form $\frac{m}{q^{n+1}}$. But not all numbers of this form have a finite representation, e.g., $x = 11/8$ from Example 8 has eventually periodic representation $\cdots 111111222$. To understand this better, we rewrite Equation (1) from the definition of the MD algorithm for this special case when $t = q^{n-1}$ as follows:

$$ps_{i+1} = qs_i - a_i q^{n+1}.$$

Now, employing the trivial fact that s_0 is a multiple of q^0, we can see that s_1 is a multiple of q^1. In this way we can prove that s_i is a multiple of q^i if $i < n+1$, and that s_i is a multiple of q^{n+1} otherwise. It implies that after at most $n+1$ steps of the MD algorithm we obtain a integer on the left side of (1). As we know from the previous subsection, if this integer is nonnegative, the sequence ends in infinitely many zeros, if it is negative, the sequence is eventually periodic with period 1. And this idea is a stepping stone for the proof of this lemma:

Lemma 10. *Denote* $F(L) = \{k \in \mathbb{N} \mid \frac{k}{q^L}$ *has infinite* $\frac{1}{q}\frac{p}{q}$*-representation*$\}$. *Then* $F(1) = \emptyset$ *and*

$$F(L+1) = \left\{\left\{-kp + mq^L \mid k > 1, m \in \mathcal{A}_p\right\} \cap \mathbb{N}\right\} \cup \left\{pk + mq^L \mid k \in F(L), m \in \mathcal{A}_p\right\}.$$

Since the recursive relation for $F(L)$ is a bit tricky, let us consider an example.

Example 11. *Let* $p = 3, q = 2$. *Then* $F(1) = \emptyset$ *and*

$$\begin{aligned}
F(2) &= \{-3 + 2 * 2\} = \{1\}, \quad \textit{indeed } \frac{1}{4} \textit{ has an infinite representation,}\\
F(3) &= \{-6 + 2 * 4, -3 + 1 * 4, -3 + 2 * 4\} \cup \{1 * 3 + 0 * 4, 1 * 3 + 1 * 4, 1 * 3 + 2 * 4\}\\
&= \{2, 1, 5, 3, 7, 11\},\\
F(4) &= \{1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 13, 14, 15, 17, 19, 21, 22, 23, 25, 29, 31, 33, 37, 41, 49\}.
\end{aligned}$$

2.4 Representation of r-adic numbers

In this subsection, we will consider the general case of $\frac{1}{q}\frac{p}{q}$-representations of r-adic numbers, r prime. We again do not assume that p is a prime number, only that $p > q \geq 1$ are co-prime. First question we will answer is the question on the number of such representations of a given $x \in \mathbb{Q}_r$. To be able to do so, we need to know some simple facts, for proofs see again [3].

Lemma 12. *Let* r *be a prime, then* $|\ |_r$ *is* ultrametric, *i.e., for all* $x, y \in \mathbb{Q}_r$ *it holds that*

$$|x + y|_r \leq \max\{|x|_r, |y|_r\}.$$

Lemma 13. *Let* r *be a prime,* $x \in \mathbb{Q}_r$ *such that* $|x|_r \leq 1$, *and* $n \in \mathbb{N}$. *Then there exists a unique* $\alpha_n \in \{0, 1, \ldots, r^n - 1\}$ *such that*

$$|x - \alpha_n|_r \leq r^{-n}.$$

It is a direct consequence of the construction of the r-adic absolute value that the infinite series of the form $\sum \frac{a_i}{q}\left(\frac{p}{q}\right)^i$ converges only in $\mathbb{Q}_r$ where r is a prime factor of p. That is why we will restrict ourselves to this case.

Lemma 14. *Let* r *be a prime factor of* p *with multiplicity*[3] i *and let* $x \in \mathbb{Q}_r$. *If* $\mathbf{a} = \cdots a_1 a_0 a_{-1} \cdots a_{-\ell_0}, a_i \in \mathcal{A}_p$, *such that*

$$x = \sum_{k=-\ell_0}^{\infty} \frac{a_k}{q}\left(\frac{p}{q}\right)^k,$$

then we have for all integers $n \geq -\ell_0$

$$\left| x - \sum_{k=-\ell_0}^{n} \frac{a_k}{q}\left(\frac{p}{q}\right)^k \right|_r \leq r^{-(n+1)i}.$$

[3] It means that i is the maximal integer such that r^i divides p.

Proof. We have

$$x - \sum_{k=-\ell_0}^{n} \frac{a_k}{q}\left(\frac{p}{q}\right)^k = \sum_{k=n+1}^{\infty} \frac{a_k}{q}\left(\frac{p}{q}\right)^k.$$

Denote

$$B_{n,m} = \sum_{k=n+1}^{n+m} \frac{a_k}{q}\left(\frac{p}{q}\right)^k, \qquad \text{with } n \geq \ell_0, m > 0.$$

For all such integers n and m we get by hypothesis on r and i and by the fact that r-adic absolute value is ultrametric:

$$|B_{n,m}|_r \leq \max_{k=n+1,\dots,n+m} \frac{a_k}{q}\left(\frac{p}{q}\right)^k \leq r^{-(n+1)i}.$$

Since it is true for all m, we are done.

□

So we know that it is enough to consider only those $\frac{1}{q}\frac{p}{q}$-representation of $x \in \mathbb{Q}_r$ whose "speed of convergence" is proportional to r^{-i}. All such representations are described by the following proposition.

Proposition 15. *Let r be a prime factor of p with multiplicity i and let $x \in \mathbb{Q}_r$, $|x|_r \leq 1$. Then:*

(i) There exist uncountably many $\frac{1}{q}\frac{p}{q}$-representations $\mathbf{a} = \cdots a_2 a_1 a_0, a_i \in \mathcal{A}$, of x such that

$$\left| x - \sum_{k=0}^{n} \frac{a_k}{q}\left(\frac{p}{q}\right)^k \right|_r \leq r^{-(n+1)i}. \tag{2}$$

Each of these representations is determined by an infinite sequence $(m_j)_{j\geq 0}$, $m_j \in \{0, 1, \dots, \bar{r} - 1\}$, where $p = r^i \bar{r}$.

(ii) If p is a prime number, thus $r = p$ and $i = 1$, then x has a unique $\frac{1}{q}\frac{p}{q}$-representation satisfying (2).

Proof. If $|x|_r \leq 1$, then $|qx|_r \leq 1$ as well. By Lemma 13 we know that there exits a unique $u_0 \in \{0, 1, \dots, r^i - 1\}$ such that

$$|qx - u_0|_r =\leq r^{-i}.$$

Since the r-adic absolute value is ultrametric, we have for all $m \in \mathbb{N}$

$$|qx - (u_0 + mr^i)|_r \leq \max\{|qx - u_0|_r, |mr^i)|_r\} \leq r^{-i}.$$

Put $a_0 = u_0 + m_0 r^i$, for some $m_0 \in \{0, 1, \dots, \bar{r} - 1\}$, then

$$|qx - a_0|_r = |x - \frac{a_0}{q}| \leq r^{-i} \qquad \text{with } a_0 \in \mathcal{A}_p.$$

The integers a_0 of this form are the only integers of $\mathcal{A}_p$ satisfying this inequality.

Now, since $|1/p|_r = r^i$, by multiplying the inequality by $|1/p|_r$ we get

$$\left|\frac{x - \frac{a_0}{q}}{p}\right|_r \leq 1$$

and so, as above, we have unique $u_1 \in \{0, 1, \ldots, r^i - 1\}$, arbitrary $m_1 \in \{0, 1, \ldots, \bar{r} - 1\}$ and $a_1 = u_1 + m_1 r^i$ such that

$$\left|q^2 \frac{x - \frac{a_0}{q}}{p} - u_1\right|_r \leq r^{-i}.$$

Multiplying by $|p/q^2|_r = r^{-i}$ yields

$$\left|x - \frac{a_0}{q} - \frac{a_1}{q}\frac{p}{q}\right|_r \leq r^{-2i}.$$

In this way, after n steps we obtain

$$\left|x - \sum_{k=0}^{n} \frac{a_k}{q}\left(\frac{p}{q}\right)^k\right|_r \leq r^{-(n+1)i}.$$

□

This lemma can be even generalized using a bit more sophisticated notation and considering only rational x.

Definition 16. *Let $p = r_1^{\ell_1} \cdots r_k^{\ell_k}$ be a prime factorization of p, r_j are prime numbers > 1 and $\ell_j > 0$. Let $\mathbf{y} = (y_1, \cdots, y_k) \in \{0, \ell_1\} \times \cdots \times \{0, \ell_k\} \backslash (0, 0, \ldots, 0)$, then $r^{\mathbf{y}} = r_1^{y_1} \cdots r_k^{y_k}$, $I(\mathbf{y}) = \{j \mid y_j = \ell_j\}$, and $\overline{r^{\mathbf{y}}}$ is defined by $p = r^{\mathbf{y}}\overline{r^{\mathbf{y}}}$.*

Now, for all admissible $\mathbf{y}$, if you consequently replace r by all $r_j, j \in I(\mathbf{y})$, and $\bar{r}$ by $\overline{r^{\mathbf{y}}}$, the lemma is still true, i.e., for any $\mathbf{y}$, there exists $\frac{1}{q}\frac{p}{q}$-representation of $x \in \mathbb{Q}$ which converges to x with respect to $|\ |_{r_j}$ for all $j \in I(\mathbf{y})$; moreover, the number of such representation is given by the number of sequences $(m_j)_{j \geq 0}$ with $m_j \in \{0, \ldots, \overline{r^{\mathbf{y}}} - 1\}$.

Corollary 17. *Let r be a prime number. Then $x \in \mathbb{Q}_r$ has a $\frac{1}{q}\frac{p}{q}$-representation if, and only if, r is a prime factor of p.*

In particular, there exists a unique $\frac{1}{q}\frac{p}{q}$-representation over the alphabet $\{0, 1, \ldots, r^i - 1\}$, where i is a multiplicity of r in p.

After going through the proof of Proposition 15 (or better of its generalization mention below the proof) carefully, one can come up with an algorithm returning all possible $\frac{1}{q}\frac{p}{q}$-representations of a given rational number. As this algorithm is a straightforward generalization of the MD algorithm, we call it *generalized modified division (GMD) algorithm.*

Algorithm 18. *Let* $\mathbf{y}$ *be fixed but arbitrary for a given* p *(see Definition 16) and* $x = \frac{s}{t} \in \mathbb{Q}$ *such that* t *is co-prime to* r_j *for all* $j \in I(\mathbf{y})$. *Put* $s_0 = s, t_0 = t$. *Further:*

$$\begin{aligned} t_j &= t_{j-1}\overline{r^{\mathbf{y}}} = t_0(\overline{r^{\mathbf{y}}})^j \\ q\frac{s_j}{t_j} &= \frac{s'_{j+1}}{t_j}r^{\mathbf{y}} + \frac{u_j t_j}{t_j} \qquad \textit{with } u_j \in \{0, 1, \ldots, r^{\mathbf{y}} - 1\}, \end{aligned}$$

choose $m_j \in \{0, 1, \ldots, \overline{r^{\mathbf{y}}} - 1\}$ *at random and put*

$$\begin{aligned} a_j &= u_j + m_j r^{\mathbf{y}} \\ s_{j+1} &= s'_{j+1} - m_j t_j. \end{aligned}$$

Denote the set of all possible outputs $\mathbf{a} = \cdots a_2 a_1 a_0$ *by GMD*(x).

Lemma 19. *Let* $\mathbf{y}$ *and* s/t *satisfy assumptions of the previous algorithm. Then* $\mathbf{a}$ *is* $\frac{1}{q}\frac{p}{q}$*-representation of* s/t *converging in all spaces* $\mathbb{Q}_{r_j}, j \in I(\mathbf{y})$*, if, and only if,* $\mathbf{a} \in$ *GMD*(x).

Corollary 20. *If* t *is co-prime to all prime factors of* p*, then there exists a unique* $\frac{1}{q}\frac{p}{q}$*-representation of* s/t *which represents* s/t *in all spaces* $\mathbb{Q}_{r_j}$, $j = 1, \ldots, k$. *This representation is equal to* $< s/t >_{\frac{1}{q}\frac{p}{q}}$ *the output of the original MD algorithm.*

Example 21. *Let* $p = 12 = 2^2 * 3, q = 7$. *Then GMD(1) for* $\mathbf{y} = (2, 0)$ *contains:* $\cdots 0123331000321113313, \cdots 6744645665754667767$*, these representations (aperiodic!) converges to 1 with respect to* $|\ |_2$*.The first one is the unique representation over the alphabet* $\{0, 1, 2, 3 = 2^2 - 1\}$.

The (finite) representation $< 1 >_{\frac{1}{q}\frac{p}{q}}$ *corresponds to* $\mathbf{y} = (2, 1)$ *and is equal to* $7(= \cdots 0007)$.

Note that even positive integer can have infinite aperiodic $\frac{1}{q}\frac{p}{q}$-representation! In fact, the following holds:

Lemma 22. *Let* $x \in \mathbb{Q}_r, r$ *a prime factor of* p. *Then a* $\frac{1}{q}\frac{p}{q}$*-representation* $\mathbf{a}$ *of* x *is eventually periodic if, and only if,* $x \in \mathbb{Q}$ *and* $\mathbf{a} =< x >_{\frac{1}{q}\frac{p}{q}}$.

3 β-expansions of r-adic numbers

In the present section we briefly summarize some consequences of what we have done so far for β-expansions of r-adic numbers.

Definition 23. *Let* $\beta \in \mathbb{Q}_r, r$ *prime, such that* $|\beta|_r \leq 1$*, i.e.,* $|\beta|_r = r^{-\ell}$ *for some* $\ell > 0$. *Define the alphabet* $\mathcal{A}_\beta = \{a \in \mathbb{N} \mid a < (|\beta|_r)^{-1}\} = \{0, 1, \ldots, r^\ell - 1\}$ *(an analogue to* $\mathcal{A}_{\lfloor\beta\rfloor}$ *in the real case). Then for a given* $x \in \mathbb{Q}_r$ *any left infinite word* $\mathbf{a}$ *over this alphabet satisfying*

$$x = \sum^{\infty} a_i \beta^i$$

is called a β-expansion of x in $\mathbb{Q}_r$.

After a slight modification of the proof of Proposition 15 we can get the same statement for $\frac{p}{q}$-representation. Hence, as its corollary, we have:

Proposition 24. *Let $\beta \in \mathbb{Q}_r, r$ prime, such that $|\beta|_r = r^{-\ell}$ for $\ell > 0$. Then for every $x \in \mathbb{Q}_r, |x|_r \leq 1$ there exists a unique word $\mathbf{a} = \cdots a_1 a_0$ over the alphabet $\mathcal{A}_\beta$ such that*

$$x = \sum_{i=0}^{\infty} a_i \beta^i.$$

Moreover, for all $n \in N$:

$$\left| x - \sum_{i=0}^{\infty} a_i \beta^i \right|_r \leq r^{-(n+1)\ell}.$$

Now, considering only rational $\beta = \frac{p}{q}$, we can prove an analogue of Lemma 22 for $\frac{p}{q}$-representations. The main idea is to replace equality $q\frac{s_i}{t} = p\frac{s_{i+1}}{t} + a_i$ in MD algorithm by $q\frac{s_i}{t} = p\frac{s_{i+1}}{t} + a_i q$. Then we have

$$\frac{s_0}{t} = \frac{p}{q}\frac{s_1}{t} + a_0 = \cdots = \left(\frac{p}{q}\right)^n \frac{s_n}{t} + \sum_{k=0}^{n-1} a_k \left(\frac{p}{q}\right)^k,$$

i.e., a representation of the form of $\frac{p}{q}$-representation analogous to $< s/t >_{\frac{1}{q}\frac{p}{q}}$. It is possible to prove that if $\frac{p}{q}$-expansion of $x \in \mathbb{Q}_r$ is eventually periodic, than $x \in \mathbb{Q}$, but the reverse implication is not true (see GMD(1) in Example 21, analogous results can be proved for $\frac{p}{q}$-representations). It would be also interesting to study β-expansions for not-rational β, e.g., for r-adic analogue of $\sqrt{2}$ (the solution of $x^2 = 2$ in $\mathbb{Q}_r$). So far, we have no idea how to attack such a problem.

References

[1] S. Akiyama, C. Frougny, and J. Sakarovitch. *Powers of rationals modulo 1 and rational base number systems.* Israel J. Math **168** (2008), 53–91.

[2] P. Ambrož. *Algebraic and combinatorial properties of non-standard numeration systems.* PhD thesis, Université Paris VII and Czech Technical University, (2006).

[3] F. Q. Gouvêa. *p-adic numbers: an introduction.* Universitext. Springer, (1997).

[4] K. Hensel. *.ber eine neue Begr.ndung der Theorie der algebraischen Zahlen.* Jahresber. Deutsch. Math. Verein **6** (1987), 83–88.

[5] M. R. Murty. *Introduction to p-adic analytic number theory.* American Mathematical Society, (2002).

[6] A. Rényi. *Representations for real numbers and their ergodic properties.* Acta Math. Acad. Sci. Hungar. **8** (1957), 477–493.

Different Approaches of Study Direct Equivalence Characterization*

Václav Kratochvíl

4th year of PGS, email: velorex@utia.cas.cz
Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, CTU in Prague
advisor: Radim Jiroušek, Institute of Information Theory and Automation, ASCR

Abstract. Structure of each Compositional model can be visualized by a tool called persegram. Every persegram over a finite non-empty set of variables N induces an independence model over N, which is a list of conditional independence statements over N. The *equivalence problem* is how to characterize (in graphical terms) whether all independence statements in the model induced by persegram $\mathcal{P}$ are in the model induced by a second persegram $\mathcal{P}'$ and vice versa. This problem can be solved either by direct od indirect characterization.

In this paper we give the motivation and introduction for direct characterization of equivalence. We have found some necessary invariant properties among equivalent persegrams that have to be satisfied. The opposite implication (whether these properties are sufficient too) is still missing. However, a very powerful tool to recognize non-equivalent persegrams is introduced in the paper.

Abstrakt. Každý kompozicionální model indukuje svoji strukturou množinu nezávislostí - tzv. indukovaný nezávislostní model. Struktura kompozicionálního modelu se obvykle znázorňuje pomocí persegramu. Proto říkáme že nezávislostní model je indukován persegramem. Rozhodnout, zda dané dva persegramy indukují stejný nezávislostní model není jednoduché. Tento problém se označuje jako *problém ekvivalence*. Případné persegramy jako ekvivalentní. Řešením je přímá nebo nepřímá charakterizace.

Tento článek se zabývá přímou charakterizací. Jsou publikovány některé nutné podmínky (invariantní vlastnosti) ekvivalentních persegramů. Přestože důkaz zda jsou podmínky postačující stále chybí, představují publikované invariantní podmínky silný nástroj na řešení problému ekvivalence.

The ability to represent and process multidimensional probability distributions is a necessary condition for the application of probabilistic methods in Artificial Intelligence. Among the most popular approaches are the methods based on Graphical Markov Models, e.g., Bayesian Networks. The Compositional models are an alternative approach to Graphical Markov Models. These models are generated by a sequence (generating sequence) of low-dimensional distributions, which, composed together, create a distribution - the so called *Compositional model*. Moreover, while a model is composed together, a system of (un)conditional independencies is simultaneously introduced by the structure of the generating sequence.

*The research was partially supported by Ministry of Education of the Czech Republic under grant no. 2C06019, 1M0572, and by Czech Science Foundation under grants no. ICC/08/E010 and 201/09/1891.

The structure can be visualized by a tool called *persegram* and one can read induced independencies directly using this tool. That is why we can say that every persegram over a finite non-empty set of variables N induces an *independence model* over N - a list of conditional independence statements over N. The *equivalence problem* is how to characterize (in graphical terms) whether all independence statements in the model induced by persegram $\mathcal{P}$ are also in the independence model induced by a second persegram $\mathcal{P}'$ and vice versa.

1 Notation and Basic Properties

Throughout the paper the symbol N will denote a non-empty set of finite-valued *variables*. From the next chapter on, variables will be represented by markers of a persegram. All probability distributions of this variables will be denoted by Greek letters (usually π, κ); thus for $K \subset N$, we consider a distribution (a probability measure over K) $\pi(K)$ which is defined for variables K. When several distributions will be considered, we shall distinguish them by indices. For a probability distribution $\pi(K)$ and $U \subset K$ we will consider a *marginal distribution* $\pi(U)$.

The following conventions will be used throughout the paper. Given sets $K, L \subset N$ the juxtaposition KL will denote their union $K \cup L$. The following symbols will be reserved for special subsets of N: K, R, S. The symbol U, V, W, Z will be used for general subsets of N. The symbol $|U|$ will be used to denote the number of elements of a finite set U, that is, its *cardinality.* u, v, w, z denotes variables as well as singletons $\{u\}, \ldots$

Independence and dependence statements over N correspond to special *disjoint triples* over N. The symbol $\langle U, V|Z\rangle$ denotes a triplet of pairwise disjoint subsets U, V, Z of N. This notations anticipates the intended meaning: the set of variables U is conditionally independent or dependent of the set of variables V given the set of variables Z. This is why the third set Z is separated by a straight line: it has a special meaning of the conditioning set. The symbol $\mathcal{T}(N)$ will denote the class of all disjoint triplets over N:

$$\mathcal{T}(N) = \{\langle U, V|Z\rangle; U, V, Z \subseteq N \quad U \cap V = V \cap Z = Z \cap U = \emptyset\}$$

To describe how to compose low-dimensional distributions to get a distribution of a higher dimension we use the following operator of composition.

Definition 1.1. For arbitrary two distributions $\pi(K)$ and $\kappa(L)$ their *composition* is given by the formula

$$\pi(K) \triangleright \kappa(L) = \begin{cases} \frac{\pi(K)\kappa(L)}{\kappa(K\cap L)} & \text{if } \pi^{\downarrow K\cap L} \ll \kappa^{\downarrow K\cap L}, \\ \text{undefined} & \text{otherwise,} \end{cases}$$

where the symbol $\pi(M) \ll \kappa(M)$ denotes that $\pi(M)$ is *dominated* by $\kappa(M)$, which means (in the considered finite setting)

$$\forall x \in \times_{j\in M}\mathbf{X}_j; (\kappa(x) = 0 \Longrightarrow \pi(x) = 0).$$

The result of the composition (if defined) is a new distribution. We can iteratively repeat the process of composition to obtain a multidimensional distribution - a model approximating the original distribution with corresponding marginals. That is why the multidimensional distribution (and the whole theory as well) is called *Compositional model*. To describe such a model it is sufficient to introduce an ordered system of low-dimensional distributions $\pi_1, \pi_2, \ldots, \pi_n$. If all compositions are defined, we call this ordered system a *generating sequence*.

From now on, we consider generating sequence $\pi_1(K_1), \pi_2(K_2), \ldots, \pi_n(K_n)$ which defines a distribution (where the operator $\triangleright$ is applied from left to right)

$$\pi_1(K_1) \triangleright \pi_2(K_2) \triangleright \ldots \triangleright \pi_n(K_n).$$

Therefore, whenever distribution π_i is used, we assume it is defined for variables K_i. In addition, each set K_i can be divided into two disjoint parts. We denote them R_i and S_i with the following sense:

$$R_i = K_i \backslash (K_1 \cup \ldots \cup K_{i-1}), S_i = K_i \cap (K_1 \cup \ldots \cup K_{i-1})$$

.

R_i denotes variables from K_i with the first appeared with respect to the sequence (meaning from left to right). S_i denotes the already used.

1.1 Graphical concepts

It is well-known that one can read conditional independence relations of a Bayesian network from its graph. A similar technique is used in compositional models. An appropriate tool for this is a *persegram*. Persegram is used to visualize the structure of a compositional model and is defined bellow. The example of persegram can be found in the Example 1.5.

Definition 1.2. Persegram $\mathcal{P}$ *of a generating sequence is a table in which rows correspond to variables (in an arbitrary order) and columns to low-dimensional distributions; ordering of the columns corresponds to the generating sequence ordering. A position in the table is marked if the respective distribution is defined for the corresponding variable. Markers for the first occurrence of each variable (i.e., the leftmost markers in rows) are squares (we call them* box-markers*) and for other occurrences there are* bullets.

Since the markers in the i-th column represent variables K_i, we denote markers in i-th column as K_i. Box-markers in i-th column of $\mathcal{P}$ are denoted like R_i and bullets like S_i. $K_i = R_i \cup S_i$. The symbol $|\mathcal{P}|$ will be used to denote the number of columns of $\mathcal{P}$, that is, its *length*. This notation is purposely in accordance with notation of variable sets in generating sequences to simplify readability and lucidity of the text.

Persegrams are usually denoted by $\mathcal{P}$ and if it is not specified otherwise, $\mathcal{P}$ corresponds to the generating sequence $\pi_1(K_1), \ldots, \pi_n(K_n)$ where $K_1 \cup \ldots \cup K_n = N$. We say that $\mathcal{P}$ is *defined over* N. (i.e. $\mathcal{P}$ over N has n columns with markers $K_1, \ldots, K_n$ where $K_1 \cup \ldots \cup K_n = N$.)

To simplify the notation we will use the following symbol: Let $\mathcal{P}$ be a persegram over N. We introduce a function $][_{\mathcal{P}}\colon N \rightarrow \mathbb{N}$, which for every variable $u \in N$ returns the index of set K_i with the first appearance of u in the persegram $\mathcal{P}$. Due to the previously established notation can be said that $K_{]u[_\mathcal{P}}$ is a column K_i where $u \in R_i$. In other words: $]u[_\mathcal{P}= i : u \in R_i$.

Definition 1.3. *Let $\mathcal{P}$ be a persegram over N and $\preceq_\mathcal{P}$ a binary relation. For arbitrary $u, v \in N$ we denote $u \preceq_\mathcal{P} v$ if $]u[_\mathcal{P} \leq]v[_\mathcal{P}$. Moreover we introduce the relation $\prec_\mathcal{P}$: $u \prec_\mathcal{P} v \Leftrightarrow u \preceq_\mathcal{P} v$ AND $v \npreceq_\mathcal{P} u$.*

The following convention will be used throughout the paper: Given variables $u, v, w \in N$ and $\mathcal{P}$ over N, the term $u, v \prec_\mathcal{P} w$ denotes that $u \prec_\mathcal{P} w$ and $v \prec_\mathcal{P} w$. The symbol $\mathcal{P}$ may be omitted, if the content is clear.

1.2 Conditional independence

Conditional independence statements over N induced by the structure of Compositional model can be read from its persegram. Such independence is indicated by the absence of a *trail connecting or avoiding relevant markers.* It is defined below.

Definition 1.4. *Consider a persegram over N and a subset $Z \subset N$. A sequence of markers $m_0, \ldots, m_t$ is called a* Z-avoiding trail *that connects m_0 and m_t if it meets the following 4 conditions:*

1. *for each $s = 1, \ldots, t$ a couple (m_{s-1}, m_s) is in the same row (i.e., horizontal connection) or in the same column (vertical connection);*
2. *each vertical connection must be adjacent to a box-marker (one of the markers is a box-marker);*
3. *no horizontal connection corresponds to a variable from Z;*
4. *vertical and horizontal connections regularly alternate with the following possible exception: two vertical connections may be in direct succession if their common adjacent marker is a box-marker of a variable from Z;*

If a Z-avoiding trail connects two-box markers corresponding to variables u and v, we also say that these variables are connected by a Z-avoiding trail. *Suppose $\langle U, V|Z\rangle \in \mathcal{T}(N)$ is a disjoint triplet over N. One says that U and V are conditionally dependent by Z, written $U \not\perp\!\!\!\perp V|Z[\mathcal{P}]$, if there exists a Z-avoiding trail between variable $u \in U$ and variable $v \in V$ in $\mathcal{P}$. In the opposite case one says that U and V are conditionally independent by Z in $\mathcal{P}$, written $U \perp\!\!\!\perp V|Z[\mathcal{P}]$. We also say that $\langle U, V|Z\rangle$ is* represented *in $\mathcal{P}$. The induced independence model $\mathcal{I}(\mathcal{P})$ and the induced dependence model $\mathcal{D}(\mathcal{P})$ are defined as follows:*

$$\mathcal{I}(\mathcal{P}) = \{\langle U, V|Z\rangle \in \mathcal{T}(N); U \perp\!\!\!\perp V|Z[\mathcal{P}]\}$$

$$\mathcal{D}(\mathcal{P}) = \{\langle U, V|Z\rangle \in \mathcal{T}(N); U \not\perp\!\!\!\perp V|Z[\mathcal{P}]\}$$

Example 1.5. *Consider persegram from Figures 1 and 2.*

In Figure 1 a $\emptyset$-avoiding trail is depicted. Therefore $u \not\perp\!\!\!\perp z|\emptyset$. Moreover, one can replace $\emptyset$ by any subset of $\{v, w, x, y\}$ which is avoiding Z as well. In Figure 2, there is depicted another trail connecting u and x. Therefore $u \not\perp\!\!\!\perp x|z$. On the contrary to Figure 1, one can not replace z by any other variable except v. Otherwise, the condition 3. from the Definition 1.4 will be corrupted. (i.e. $u \perp\!\!\!\perp x|y[\mathcal{P}]$ for example)

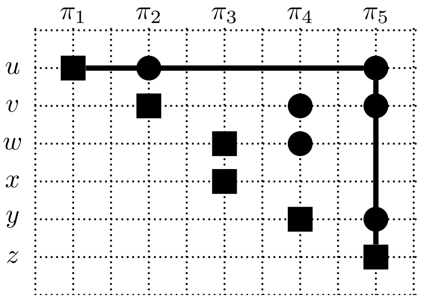


Figure 1: $\mathcal{P} : u \not\perp\!\!\!\perp z|\emptyset, u \not\perp\!\!\!\perp z|v$

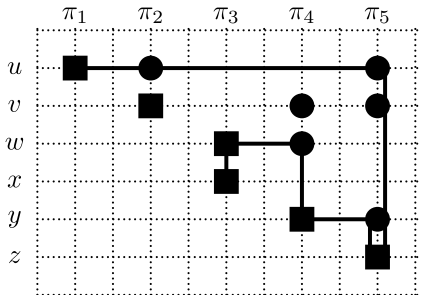


Figure 2: $\mathcal{P} : u \not\perp\!\!\!\perp x|z$

The following specific notation for certain composite dependence statements will be useful. Given a persegram $\mathcal{P}$ over N, distinct variables $u, v \in N$ and disjoint set $U \subseteq N \setminus \{u, v\}$ the symbol $u \not\perp\!\!\!\perp v| + U[\mathcal{P}]$ will be interpreted as the condition

$$u \not\perp\!\!\!\perp v| + U[\mathcal{P}] \equiv \forall W \text{ such that } U \subseteq W \subseteq N \setminus \{u, v\} \text{ one has } u \not\perp\!\!\!\perp v|W[\mathcal{P}].$$

In words, u and v are (conditionally) dependent in $\mathcal{P}$ given any superset of U. If U is empty we write $*$ instead of $+\emptyset$. I.e.

$$u \not\perp\!\!\!\perp v| * [\mathcal{P}] \equiv \forall W \text{ such that } W \subseteq N \setminus \{u, v\} \; u \not\perp\!\!\!\perp v|W[\mathcal{P}].$$

We give a certain graphical characterization of composite dependence statements of this kind below.

2 Equivalence problem

By the equivalence problem we understand the problem how to recognize whether two given persegrams $\mathcal{P}, \mathcal{P}'$ over N induce the same independence model ($\mathcal{I}(\mathcal{P}) = \mathcal{I}(\mathcal{P}')$). It is of special importance to have an easy rule to recognize that two persegrams are equivalent in this sense and an easy way to convert $\mathcal{P}$ into $\mathcal{P}'$ in terms of some elementary operations on persegrams. Another very important aspect is the ability to generate all persegrams which are equivalent to a given persegram.

Definition 2.1. *Persegrams $\mathcal{P}, \mathcal{P}'$ (over the same variable set N) are called* independence equivalent*, if they induce the same independence model $\mathcal{I}(\mathcal{P}) = \mathcal{I}(\mathcal{P}')$.*

Remark 2.2. *One may easily see that the above mentioned definition could be formulated with the term of dependence model.* Persegrams $\mathcal{P}, \mathcal{P}'$ (over the same variable set N) are independence equivalent, iff $\mathcal{D}(\mathcal{P}) = \mathcal{D}(\mathcal{P}')$. *This alternative is used in most proofs primarily.*

2.1 Direct characterization

The solution of equivalence problem can be done in several ways. Some kind of *direct characterization* of equivalence follows was done in the paper [5] where we introduced two invariant properties of equivalent persegrams. Let us remind these invariant together with necessary definitions of *connection* and *ordering condition.* Proofs can be found in [5] as well.

Definition 2.3. *Let $\mathcal{P}$ be a persegram over N and $u, v \in N$ be two distinct variables, and $u \preceq_{\mathcal{P}} v$. u, v are* connected *in $\mathcal{P}$ ($u \leftrightarrow_{\mathcal{P}} v$) if $u \in K_{]v[}$. The set of all pairs $\mathcal{E}(\mathcal{P}) = \{\langle u, v\rangle : u, v \in N, u \leftrightarrow_{\mathcal{P}} v\}$ is called a* connection set *of $\mathcal{P}$.*

The following convention will be used throughout the paper: Given variables $u, v, w \in N$ and $\mathcal{P}$ over N, the term $u, v \leftrightarrow_{\mathcal{P}} w$ denotes that $u \leftrightarrow_{\mathcal{P}} w$ and $v \leftrightarrow_{\mathcal{P}} w$. The symbol $\mathcal{P}$ may be omitted, if the content is clear.

For the purpose of the following text one should realize the obvious parallel between relation $u \leftrightarrow v$ and columns order and content. This parallel is summarized in the following remark.

Remark 2.4. *Let u, v are two different variables in $\mathcal{P}$ and $u \preceq_{\mathcal{P}} v$. Then*

$$u \leftrightarrow_{\mathcal{P}} v \Leftrightarrow u \in K_{]v[}.$$

Lemma 2.5. *Let $\mathcal{P}$ be a persegram over N and $u, v \in N$ are distinct variables. Then*

$$u \leftrightarrow_{\mathcal{P}} v \Leftrightarrow u \not\perp\!\!\!\perp v \mid * \, [\mathcal{P}].$$

Definition 2.6. *Let $\mathcal{P}$ be a persegram over N. An* Ordering condition induced by $\mathcal{P}$ *is a triplet of variables $u, v, w \in N$ where $u, v \prec w$; $u, v \leftrightarrow w$; and $u \not\leftrightarrow v$ in $\mathcal{P}$. Such an induced ordering condition is denoted by $[u, v] \prec w[\mathcal{P}]$.*

Lemma 2.7. *Let $\mathcal{P}$ be a persegram over N, $u, v, w \in N$ distinct nodes. Then*

$$[u, v] \prec w[\mathcal{P}] \Leftrightarrow u \not\perp\!\!\!\perp v \mid + w[\mathcal{P}].$$

The previous lemmata show two invariant properties of equivalent persegrams. Two persegrams, if equivalent, have the same set of connections and induce the same set of ordering conditions.

Corollary 2.8. *Let $\mathcal{P}, \mathcal{P}'$ be two persegrams over N. If $\mathcal{I}(\mathcal{P}) = \mathcal{I}(\mathcal{P}')$ then $\mathcal{E}(\mathcal{P}) = \mathcal{E}(\mathcal{P}')$ and they induce the same set of Ordering conditions.*

3 Column approach

In the previous section two invariant properties were introduced. However, the condition of the same *Connections set* is not so simply verifiable on the contrary to Bayesian networks. It will be nice to transform this condition into some other condition about columns.

The following lemma gives an interesting assertion about columns with mutually connected set of variables. Basically, it is a generalization of the Remark 2.4.

Lemma 3.1. *Let $\mathcal{P}$ be a persegram over N, $U \subseteq N$ be a set of mutually connected variables in $\mathcal{P}$($\forall u, v \in U; u \leftrightarrow_{\mathcal{P}} v$). Then $\exists u \in U$ such that $U \subseteq K_{]u[}$.*

Proof. The proof is done by induction on $|U|$. The induction hypothesis for $n \geq 2$ is that the lemma holds for any U with $|U| \leq n$. It is evident for $|U| = 2$. It follows from the Definition 2.3 or from the Remark 2.4 as well.

Assume $n = |U| \geq 2$ and that the implication holds for subsets with cardinality smaller than n. Choose $u \in U$ such that all other $u' \in U$ $u \preceq_{\mathcal{P}} u'$. This choice is always possible and ensures that $u \in K_{]u'[}$ by Remark 2.4 for all $u' \in U$. Introduce U' as $U' \equiv U \setminus \{u\}$. By the induction hypothesis, $\exists u'' \in U'$ such that $U' \subseteq K_{]u'[}$. By choice of u, it is easily verified that $U \subseteq K_{]u''[}$. □

The above mentioned lemma can be further generalized.

Lemma 3.2. *Let $\mathcal{P}, \mathcal{P}'$ be two equivalent persegrams over N, $K^{\mathcal{P}}$ an arbitrary column of $\mathcal{P}$, and an arbitrary subset $U \subseteq K^{\mathcal{P}}$ with at least one box-marker $U \cap R^{\mathcal{P}} \neq \emptyset$. Then $\exists u \in U$ such that $U \subseteq K^{\mathcal{P}'}_{]u[}$ in $\mathcal{P}'$.*

Proof. One can easily divide U into two groups. Let $R \equiv U \cap R^{\mathcal{P}}$ be the part composed from box-markers and S the rest. $U = R \cup S$.

If $|S| < 2$, then the lemma is a trivial corollary of the Lemma 3.1 (By definition, variables in R are mutually connected, and every variable from S is connected with all variables from R. Since $|S| = 1$ then all variables from $U \equiv R \cup S$ are mutually connected.) Suppose $|S| >= 2$ and $M \equiv R$ is a set of mutually connected variables. Then two possibilities exist for every $s \in S$.

1. $s \leftrightarrow s'$ for all other $s' \in S$. In that case, s can be added into a set of mutually connected variables $M = M \cup \{s\}$ and by the Lemma 3.1 there exists $m \in M$ such that $M \subseteq K^{\mathcal{P}'}_{]m[}$.

2. $\exists s' \in S$ such that $s \not\leftrightarrow s'$. Then $[s, s'] \prec r; \forall r \in R$. By Corollary 2.8 and Remark 2.4 $\forall r \in R; s, s' \in K^{\mathcal{P}'}_{]r[}$. It follows from the previous step that $\exists m \in M$ such that $R \subseteq K^{\mathcal{P}'}_{]m[}$ and therefore by Remark 2.4 $R \preceq_{\mathcal{P}'} m$. Since $s, s' \prec_{\mathcal{P}'} R \preceq_{\mathcal{P}'} m$. Therefore $s, s' \prec_{\mathcal{P}'} m$. By definition of M $s, s' \leftrightarrow m$. Then by Remark 2.4 $s, s' \in K^{\mathcal{P}'}_{]m[}$.

□

One can expand the previous assertion by induction into the following corollary.

Corollary 3.3. *Let $\mathcal{P}, \mathcal{P}'$ be two equivalent persegrams over N. Then every column K of $\mathcal{P}$ with a square-marker either exists also in $\mathcal{P}'$, or it is a subset of some other column in $\mathcal{P}$ with at least one box-marker - out of K.*

Anyway there is one column which definitely exists in an equivalent persegram. It is the last column with a box-marker. It is obvious. The last column may not be a subset of any other, since there is no column after.

This lemma can also be proved without the knowledge of the previous Lemma 3.2, only on the basis of independence invariants summarized in the Corollary 2.8.

Lemma 3.4. *Let $\mathcal{P}, \mathcal{P}'$ be two equivalent persegrams over N. If the last column of $\mathcal{P}$ - $K^{\mathcal{P}}_{|\mathcal{P}|}$ contains a box-marker, then this column is contained in $\mathcal{P}'$ as well.* ($\exists i \in 1..|\mathcal{P}'|$ *such that* $K^{\mathcal{P}'}_{i} = K^{\mathcal{P}}_{|\mathcal{P}|}$)

Proof. By the assumption the last column of $\mathcal{P}$ has to contain at least one box-marker $t \in R^{\mathcal{P}}_{|\mathcal{P}|}$. Denote other variables from $K^{\mathcal{P}}_{|\mathcal{P}|} \setminus \{t\}$ by S. $S \cup \{t\} = K^{\mathcal{P}}_{|\mathcal{P}|}$.

Since both persegrams are defined over the same variables set, then $K^{\mathcal{P}'}_{]t[}$ exists. By the Corollary 2.8 $\mathcal{E}(\mathcal{P}) = \mathcal{E}(\mathcal{P}')$ and it implies that $K^{\mathcal{P}'}_{]t[} \subseteq K^{\mathcal{P}}_{|\mathcal{P}|}$. Let $C = K^{\mathcal{P}}_{|\mathcal{P}|} \setminus K^{\mathcal{P}'}_{]t[}$. If $C = \emptyset$, then $K^{\mathcal{P}}_{|\mathcal{P}|} = K^{\mathcal{P}'}_{]t[}$ and the proof is done. Suppose $C \neq \emptyset$.

By Remark 2.4 $\forall c \in C;\ t \prec_{\mathcal{P}'} c$.

Choose $c \in C$ and corresponding $K^{\mathcal{P}'}_{]c[}$ such that other $c' \in C; c' \preceq_{\mathcal{P}'} c$. This choice is always possible and ensures that by remark 2.4 $t \prec c[\mathcal{P}']$ and $\forall s \in S;\ s \preceq_{\mathcal{P}'} c$.

The next step is to observe that $S \subseteq K^{\mathcal{P}'}_{]c[}$. Indeed, suppose that $s \not\leftrightarrow_{\mathcal{P}'} c$ for some $s \in S$. Then, $s \not\leftrightarrow_{\mathcal{P}} c$ by $\mathcal{E}(\mathcal{P}) = \mathcal{E}(\mathcal{P}')$, and $s \leftrightarrow_{\mathcal{P}} t$, $c \leftrightarrow_{\mathcal{P}} t$ by definition. Since t is in last column of $\mathcal{P}$ only, one has $[s, c] \prec t[\mathcal{P}]$ and $[s, c] \prec t[\mathcal{P}']$ by Lemma 2.7. This however contradicts the choice of c where $t \prec_{\mathcal{P}'} c$. Thus necessarily $s \leftrightarrow_{\mathcal{P}'} c$.

Another observation is that $K^{\mathcal{P}'}_{c} \subseteq S \cup \{t\}$. Indeed, suppose that there exists $v \in N \setminus S$; $v \neq t$ such that $v \in K^{\mathcal{P}'}_{]c[}$. Since $v \notin S$ one has $v \not\leftrightarrow_{\mathcal{P}} t$ and therefore $v \not\leftrightarrow_{\mathcal{P}'} t$. It implies that $v \in S^{\mathcal{P}'}_{]c[}$ and therefore $v \prec_{\mathcal{P}'} c$ (otherwise, since $t \in K^{\mathcal{P}'}_{]c[}$ and $v \in R^{\mathcal{P}'}_{]c[}$ then $t \leftrightarrow_{\mathcal{P}'} v$). Thus $[v, t] \prec c[\mathcal{P}']$ implies $[v, t] \prec c[\mathcal{P}]$ by lemma 2.7. This contradict the fact $c \preceq_{\mathcal{P}} t$.

Then $K^{\mathcal{P}'}_{]c[} = K^{\mathcal{P}}_{|\mathcal{P}|}$ necessarily. □

Remark 3.5. *The box marker t was chosen randomly in the previous proof. Thence it follows $K^{\mathcal{P}'}_{]t[} \equiv K^{\mathcal{P}}_{|\mathcal{P}|}$, but also $R^{\mathcal{P}}_{|\mathcal{P}|} \subseteq R^{\mathcal{P}'}_{]t[}$. I.e. Every variable with box-marker in $K^{\mathcal{P}}_{|\mathcal{P}|}$ has a box-marker in $K^{\mathcal{P}'}_{]t[}$.*

In addition to this remark we would like to know whether $K^{\mathcal{P}'}_{]t[}$ is *the only column* in $\mathcal{P}'$ containing $R^{\mathcal{P}}_{|\mathcal{P}|}$. Suppose that $t \in K^{\mathcal{P}'}_{]x[}$ such that $x \succ_{\mathcal{P}'} t$ and therefore $x \leftrightarrow t[\mathcal{P}']$. This contradicts the fact $x \notin K^{\mathcal{P}}_{|\mathcal{P}|}$.

In the above paragraphs we supposed that the last column of $\mathcal{P}$ contains at least one box-marker.

However, the condition of the same Connections set $\mathcal{E}(\mathcal{P})$ is a little bit difficult to verify. In case of graphs (e.g. in Bayesian networks) one simply put the graphs crisscross and the result is obvious. We will appreciate some rule concerning columns in case of persegrams.

Let us extend the Lemma 3.2 and Corollary 3.3.

Remark 3.6. *Let $K^{\mathcal{P}}_{]u[} \subset K^{\mathcal{P}}_{]v[}$, then $\forall u' \in K^{\mathcal{P}}_{]u[}$ holds that $u' \prec_{\mathcal{P}} v$.*

The assertion of the above mentioned remark is very simple. If even $u \in K_{]v[}$ then $u \prec v$. Therefore $]u[<]v[$ and all variables from $K_{]u[}$ appear sooner than v.

Let us thing about the problem more further.

Let $\mathcal{P}, \mathcal{P}'$ be equivalent persegrams over N. Suppose that there is a column K with box-marker corresponding to $r \in N$ (i.e. $K \equiv K_{]r[}$) that has no corresponding column in $\mathcal{P}'$. By the Lemma 3.2 $\exists K' \in \mathcal{P}'$ such that $K \supseteq K'$ and at least one of K is a box-marker in K'. Let $U = K' \in K \setminus K$. $|U| \neq 0$ by the assumption. (See the areas of interest on Figures 3 and 4 in the Example 3.7.)

Choose $u \in U$ and the corresponding $K^{\mathcal{P}}_{]u[}$ such that other $u' \in U$ $u \preceq_{\mathcal{P}} u'$. This choice is always possible and ensures that all other $u \in U \cap K_{]u[}$ are box-markers as well.

The next step is to observe that $K \subset K^{\mathcal{P}}_{]u[}$. Indeed, since $\{u\} \cup K$ belongs to K' and at least one of them is a box-marker then by the Lemma 3.2 there is a column containing $K \cup \{u\}$ and by the Remark 3.6 u is a box-marker. Then this column coincide with $K^{\mathcal{P}}_{]u[}$.

Another observation is that $S^{\mathcal{P}}_{]u[} = K^{\mathcal{P}}_{]r[}$. Indeed, Let $V = S^{\mathcal{P}}_{]u[} \setminus K^{\mathcal{P}}_{]r[}$ and suppose $|V| \neq 0$. Then there is some $v \in V$ such that $v \in S_{]u[}$. By the Lemma 3.2 and the Remark 3.6 there is a column $K'_{]v[}$ containing all marker from $K_{]u[}$,i.e. $\{r, u, v\}$ etc. It means that $[r, v] \prec u[\mathcal{P}$. This contradicts the fact $v \succ_{\mathcal{P}'} u$.

Then $V = \emptyset$ necessarily.

Example 3.7. *This previous problem analysis is depicted on the following Figures 3 and 4. The following convention is used in the consequent figures. The symbol* $\times$ *represents marker of which we are no sure whether it is a box-marker or a bullet. The meaning of set of markers* $\boxtimes$ *in one column is that at least one of these markers is a box-marker but we do not know which one.*

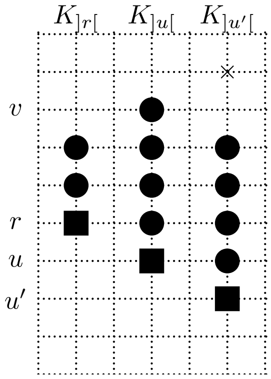


Figure 3: $\mathcal{P} : [v, r] \prec u[\mathcal{P}]$

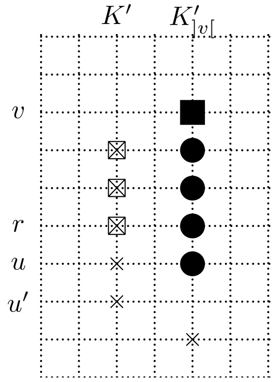


Figure 4: $\mathcal{P}' : v \succ_{\mathcal{P}'} u$

The previous paragraph extends the Lemma 3.2 in a very interesting way.

Corollary 3.8. *Let $\mathcal{P}, \mathcal{P}'$ be two equivalent persegrams over N. Then every $\forall u \in N$, either there exists a corresponding column to $K^{\mathcal{P}}_{]u[}$ in $\mathcal{P}'$ or $\exists v \in N$ such that $K^{\mathcal{P}}_{]u[} = S^{\mathcal{P}}_{]v[}$.*

Definition 3.9. *Let $\mathcal{P}$ be a persegram. Then $\mathcal{P}$ is* reduced *if there is no pair $i, j \in 1..|\mathcal{P}|$ such that $K_i = S_j$.*

Corollary 3.10. *Let $\mathcal{P}, \mathcal{P}'$ be two equivalent reduced persegrams over N. Then these persegrams consists from same columns (regardless of the markers shape).*

4 Conclusion

In this paper a short introduction into equivalence problem was given. This problem includes several sub-problems where one of them is how to recognize whether two given persegrams are equivalent "on the first sight". The partial solution to this problem is a *direct characterization* involving some invariant properties that are necessary for equivalence. Two such a properties were introduced: *Connections set* and *Ordering conditions*.

On the contrary to probability models using acyclic directed graphs (DAG) to visualize the structure, the *Connections sets* can not be so simply compared. (In case of DAGs one puts the graphs simply crisscross.) That is why we introduced the other invariant property: the *Columns set*. However, the corresponding persegrams need to be in a special *reduced* shape. Are these invariants are sufficient to decide whether two given persegrams are equivalent? Despite the promising recent research, this question remains open.

References

[1] R. Jiroušek. *Multidimensional Compositional Models.* Preprint DAR - ÚTIA 2006/4, ÚTIA AV ČR, Prague, (2006).

[2] T. Kočka, R. R. Bouckaert, M. Studený. *On the Inclusion Problem.* Research report 2010, ÚTIA AV ČR, Prague (2001).

[3] M. Studený. *O strukturách podmíněné nezávislosti.* Rukopis série přednášek. Prague (2008).

[4] V. Kratochvíl. *Equivalence Problem in Compositional Models.* Doktorandské dny 2008, Nakladatelství ČVUT, Praha, p.125-134, 2008.

[5] V. Kratochvíl. *Equivalence problem in persegrams*, The proceeding of 8th Workshop on Uncertainty Processing. In printing. (2009).

Multi-Agent Exploation in a Discrete Dynamic Environment: Dynamic Programming, Evolution Techniques, and Reactive Rules

Karel Macek

3rd year of PGS, email: karel.macek@fjfi.cvut.cz
Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, CTU in Prague
advisor: Jaromír Kukal, Department of Software Engineering in Economics, FNSPE CTU

Abstract. This work formulates the problem of renewable resources exploation in a discrete environment and provides methods that can be used for this problem. A particular instance of this problem has been examined in more detail: exact solution has been found by means of dynamic programming and compared with solutions given by designed evolution technique, using simple reactive rules.

Abstrakt. Tato práce formuluje problém těžby obnovitelných zdrojů v diskrétním prostředí a nabízí metody, které jej mohou řešit. Zároveň je zkoumána konkrétní instance tohoto problému podrobněji: pomocí dynamického programování je nalezeno exaktní řešení, které je porovnáno s navrženou evoluční metodou využívající jednoduchá reaktivní pravidla.

1 Introduction

This paper deals with with renewable resources. Renewable resources are natural resources that tend - spontaneously or with some assistance - to the original state after an external action. The modeling of renewable resource dynamics has been assessed already in several works [8] or [15] also the negotiation about them [6], [4], and other decision making issues [14]. The most important for this paper are - however - works focusing possibly most efficient usage of the resource [1],[13], [18] and causes of over-exploitation [16]. There also works addressing the action of several agents in the resources grabbing [9]. Finally, some aspects of efficient exploitation were treaten in [10].

2 Terminology

We consider the *time-line* as a totally ordered set T with *time instants* $t \in T$. The environment is represented by an arbitrary set of *places* P. In order to describe the actual state of the environment, a tupple of properties will be used. In our case, we require only *local properties*, i.e. properties related to a place. Formally, we can speak about a set of mappings $\{x_1, x_2 \dots x_M\}$ where $x_i : T \times P \to L_i \quad \forall i \in \hat{M}$. The set L_i is the *property range*, usualy a set of numbers or strings.

In the environment *agents* are placed. In this moment, it is not necessary to define an agent explictely. We can consider agents just as an arbitrary set A. The actual *position* of an agent is given by mapping $y : T \times A \to P$, i.e. given time instant and agent, the position is unique. Agents are able to make decision. The set of possible decisions is D. Actual agent's decision can be represented by mapping $a : T \times A \to D$.

Values of mappings $x_1, \dots, x_M, a, y$ are constructed incrementally as system dynamics evolves. Let *state* in time $t \in T$ be $s(t) = (t, y(t, \cdot), x(t, \cdot), a(t, \cdot)$ and s^* set of all possible states. Let s_0 be *initial state*. Let *history* be $h(t) = \bigcap_{\tau < t} S(\tau)$ and h^* set of all possible histories. Let *trajectory* be $q = \bigcap_{\tau \in T} S(\tau)$ and q^* set of all possible trajectories. The *dynamics* of the system is defined as triple of procedures $(Evolve, Move, Decide)$:

$$\begin{aligned} x_i(t, \cdot) &\leftarrow Evolve(H(t), r(t)) \qquad \forall i \in \hat{M} \qquad (1) \\ y(t, \cdot) &\leftarrow Move(H(t), r(t)) \qquad (2) \\ a(t) &\leftarrow Decide(H(t), r(t)) \qquad (3) \end{aligned}$$

where $r(t)$ is taken random from $[0,1]^{R(t)}$ and $R(t)$ is number of random values required for time instant t.

We will use following notation:

1. $\mathbb{a} \in D^{|A|,|T|}$ is a *strategy*, i.e. for all agents and all time instants.

2. $\mathbb{x}(t) \in L_1^{|P|} \times \dots \times L_M^{|P|}$ is the actual *environment state*, i.e. all local values considered.

3. $\mathbb{y}(t) \in P^{|A|}$ are actual agents' positions.

All these three procedures define the system as such. We will examine them in more detail.

Procedure *Evolve* models the behavior of the environment. It may given by differential equations, celuar automata rules etc. Procedure *Evolve* involves also impacts of decision made before in the past. The result of this procedure may be stochastic, i.e. the history and the procedure determine only a probability distribution of new local states from which new state is drawn. Procedure *Move* is very similar to the procedure *Evolve*.

Procedure *Decide* determines next agents' actions. These actions reflects the history and both actual positions and local states. Procedure *Decide* strives to solve a decision making problem. The class of possible problems is wide. However, all problems has one or several criteria that are to be optimized. These criteria depend on the entire trajectory in a given horizon $h(t_{\max})$ that may be finite or infinite which is less typical. All information about the history is not always in place. Agents work under uncertainity. Procedure *Decide* is not an action. The result of decision is given by *Move* and *Evolve*. A part of decision may not impact the system, but only change agent's inner state.

Next, we will introduce the preference and the objective function The *preference* of agent $i \in A$ is defined as a partial ordering $\prec_i$ on q^*. The objective function *objective function* is a function $f : A \times q^* \to \mathbb{R}$ such that $\forall q_1, q_2 \in q^* \forall i \in A \quad f(i, q_1) > f(i, q_2) \equiv q_1 \prec_i q_2$. Of course, there may be more objective functions in place, e.g. for different criteria, or for each agent etc. The combination of them is an interesting branch of decision making. In our case, we will work with a signle objective function.

Finally, a *multi-agent dynamic system* be $MADS = (P, A, T, s(0), Evolve, Move, Decide, f)$, where P are places, A are agents, T is time, s_0 an initial state, f is an objective function and $(Evolve, Move, Decide)$ are dynamics. Given a $MADS$, *multi-agent dynamic decision problem* is task to design function *Decide* so the expected value of f are maximal.

Now, we will introduce a specific $MADS$ and a specific problem that will be treaten in next parts of this work.

The *Discrete exploation of renewable resources system* is a multi-agent dynamic system $DERRS = (E, A, T, s_0, Evolve, Move, f)$ where

- P is a finite set
- A is a finite set
- $T = \{1, 2, \ldots, t_{\max}\}$
- $Evolve(H(t))$ is defined by procedures $NaturalEvolve$ and $Exploitation$ so that $x(t,e) = NaturalEvolve(x(t-1,\cdot)) - Exploation(t,e)$ where $Exploation(t,e) = NaturalEvolve(x(t-1,\cdot))$ if $\exists i \in A \quad y(t,i) = e$ and $Explotion(t,e) = 0$ otherwise.
- $a(t,\cdot) = Decide(H(t), r(t))$ is restriced by accessibility of places $C \subset E \times E$, i.e. $a(t,i) \in \{y' | (y(t,i), y') \in C\}$
- $y(t,\cdot) Move(H(t), r(t)) = Decide(t-1)$, i.e. the agent moves to a place for which he decided before.
- $\forall i \in A f(i,Q) = \sum_{t=1}^{t_{\max}} \sum_{e \in E} Exploitation(t,e)$

3 Dynamic programming and exact solution

Above formulated problem may be interpreted as a problem of dynamic programming which stands for a well established reasearch field [5]. The dynamic programming solution will provide exact optimal solution, but it requires much computational time. This solution has been designed and implemented so a comparison for heuristic methods is available.

Basic idea of DP approach is to solve from the final time t_{max}. Since decisions have impact always in the next time instant, for this time no decision is made. At the instant before, i.e. $t_{max} - 1$, the decision is made so the sum of exploation in the last instant is maximal etc. Thus, all configurations are treaten for each time horizon. The number of configurations grows linearly (each time horizon is solved individualy), but it is huge. In case of a very simple example defined below, the number of possible configurations is giant: $5^9 \cdot 9^2 = 158,203,125$ where 5 is number of possible values of $\varkappa$, 9 is cardiality of $\varkappa$, and finally 2 is number of agents. Therefore, we will consider all ways from beginning till the end. In this case, however, we will probably evaluate one situation more times and the solution will grow exponentially[1].

[1] Of course, an improvement can be done by saving so far known strategies. This will ensure the linear grow. However, this was not implemented yet

Let us describe the algorithm formally. Algorithm (1) shows the pseudocode of procedure Evaluate that is called recursively. This procedure calculates maximal gain for given x and y if there remain r time instants. First of all, procedure *Evolve* is called which is divided into three subprocedures *NaturalEvolve*, *Exploited*, and *Exploation*. The *Exploited* procedure returns gains. Afterwards, if there are some next steps to be considered, all possible decisions are tested: the optimal one is found recursivelly. The

Algorithm 1 Dynamic Programming Procedure

```
procedure EVALUATE($\mathbb{y}(t), \mathbb{x}(t), r$)    ▷ Returns maximal $f$ as sum of future gain $v$ and actual gain $g$
    $v \leftarrow -\infty$
    $\mathbb{x}(t+1) \leftarrow$ NATURALEVOLVE($\mathbb{x}(t)$)
    $g \leftarrow$ EXPLOITED($\mathbb{x}(t+1), \mathbb{y}(t)$)
    $\mathbb{x}(t+1) \leftarrow$ EXPLOITION($\mathbb{x}(t+1), \mathbb{y}(t)$)
    $r_{\text{next}} \leftarrow r - 1$
    if $r_{\text{next}} \neq 0$ then
        for all $\tilde{\mathbb{a}}(t) \in$ POSSIBLENEXTDECISION($C, \mathbb{y}(t)$) do
            $\tilde{\mathbb{y}}(t+1) \leftarrow \tilde{\mathbb{a}}(t)$
            $v_{\text{next}} =$ EVALUATE($\tilde{\mathbb{y}}(t+1), \mathbb{x}^{(}t+1), r_{\text{next}}$)
            $\tilde{v} \leftarrow v_{\text{next}}$
            if $\tilde{v} > v$ then
                $v \leftarrow \tilde{v}$
                $\mathbb{a}(t) = \tilde{\mathbb{a}}(t)$
            end if
        end for
    else
        $v \leftarrow 0$
    end if
    return $v + g$
end procedure
```

DP solution provides guaranteed optimal results, but the time complexity is very high. If C would define a k-regular graph, then there would be $(k \cdot n_a)^{t_{\max}}$ possibilities to be tested. This limits the applicability of the algorithm very significantly.

Possible improvement can be reached by: symetry, saving known calculations, ie Memory vs Speed.

4 Reactive Solutions

Alterantive way towards an satisfactory solution are simple rules. The discussion about thumb rules, usual for reactive agents, and dynamic programing in [7]. Such reactive rules provide usualy a suboptimal solution, but somehow good in average. This solution is provided in a moderate time and is robust, i.e. if the system is changed unexpectly, e.g. a new agent comes, the result is yet relatively good. More about heuristic decisions in dynamic programming was threaten in [11].

This section's aim is to formulate some simple approximative ways how to instruct agents in above defined DERRS.

Random Walk - agents walk through the graph randomly. At each moment, the agent draws from neigbohrs one uniformly randomly and goes in this direction.

Greedy Crowd - agents strive to exploit maximum in the next time instant. At each moment, the agent tests local properties of the neigborhood and does to the neighbohr with the maximum.

Greedy Lions and Hyenas If more neigbohrs have the same value, the agent prefers minimal index of the agent. Agents make their decisions sequentially. Therefore, if an agent is already decided to go somewhere, other greedy agents, not decided so far, wont go to this place if they have another possibility.

Aroma Tracking is a modification of both greedy approaches. The aroma of a place $e \in E$ equals to weighed average of material in its neighborhood. The agents select next position in a greedy way.

Global Pheromone Averse is also a modification of both greedy approaches. If an agent enters a field, a pheromone trail is left (some amount of pheromone is added). However, each time instant some pheromone evaporates (the amount of pheromone is multiplied by a constant from (01) interval). The agent chooses a neighbor with minimal pheromone.

Friend's Pheromone Tracking - agents leave individual pheromone trails in the way described above. Each agent i has one friend agent j that is tracked, i.e. i selects the neighbor where the pheromone trail of j is maximal.

Short Term Planning - agents will plan their actions so they are optimal in a short term horizon so the planning is possible. After each action, the planning is run again. For time horizon $t_{\max} = 0$ we can call Greedy Commando.

Of course, these simple approaches can be combined in arbitrary way. Each agent may have more rules. These rules can be combined, e.g. aroma and friend tracking where both imputs have own weights. Or, the actual rule can be chosen randomly. This sampling must not necessarily from the uniform distribution. The distribution can be also updated, e.g. by reinforcement learning.

If there are more agents in the system, each of them can have other rules, e.g. one agent A can combine aroma and friend's pheromone trackings while another agent B - the friend of A - can combine random walk with a greedy approach.

5 Evolutionary Solutions

Other way to find optimal control strategy is the evolutionary algorithms. They iterative heuristics which search optimal solution in the input space working with sereval points (population) in the input space [3].

Application of evolutionary algorithms in dynamic decision problems is nothing new [19] since discrete dynamic programming stands for a combinatorical problem [20].

The most simple approach in this way is random shooting. Here, random strategy is generated and compared with the so far best known strategy. For the random strategies, relation C is used so the correctness of generated strategies is ensured. Advanced techniques work with usual evolutionary operations.

Initialization - strategies are generated randomly (random walk).

Selection - $\rho_{\text{top}} \in [0, 1]$ percent of the best individuals are copied, $\rho_{\text{sample}} \in [0, 1]$ percent of individuals are re-sampled randomly. The rest is used for evolutionary operation. Two individuals are selected for the crossover. The population is ordered by the fitness, i.e. value how is which strategy successful. The probability of selection is proportional to the ranking. This ranking is used for the sampling instead of the fitness values because they will very probably tend to be similar. Thus, the local minima problem can be avoided.

New sub-strategies - at the beginning, the strategies are generated via random walk. For cross-over and mutation, however, more advanced approach is used. The strategy is generated piecewise. First, the generating method is chosen randomly (random walk, greedy lions and hyenas, short term planning) for the next part of strategy. Afterwards, the length of the part is determined randomly as well. Finally, the sub-strategy is generated by given method and returns also final state $\mathbb{x}$ that is used for next generation.

Cross-over - two strategies $\mathbb{a}^1, \mathbb{a}^2$ are merged. Usual cross-over is often not usefull in specific problems [17]. It is also in this case because two parts of two different strategies may not be connected in terms of the relation C. Therefore, the first part from the parent $\mathbb{a}^1$ is taken (the length of this part is random). If the part from the parent $\mathbb{a}^2$ follows respecting C, the part is mergered. Otherwise, next decisions are generated as described above from the last decision until they match the parent $\mathbb{a}^2$.

Mutation - is practically identical with the cross-over. First part is taken from the individual, the rest is generated as a new substrategy described above.

Resetting - sometimes, the population is infected by a local extreme and it is difficult to move. Therefore, if the so far best known solution was not improved too long, the process is reset. The condition is $t_{\text{li}}/t > \alpha$ where $\alpha \in (0, 1)$ and t_{li} is the time without improvement. Thus, the longer run of the algorithm, the longer trial is provided until next reset.

6 Results of experiments

For demonstrative and testing purposed, an instance of the above mentioned problem was formulated:

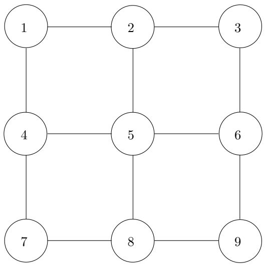


Figure 1: Set of places P in the environment and accessibility relation C used for the demonstration example.

$t_{\max}$	1	2	3	4	5	6	7	8	9	10	11
f_{opt}	4	10	18	28	37	45	54	61	70	78	87
$f_{\text{opt}}/t_{\max}$	4	5	6	7	7.4	7.5	7.714	7.625	7.777	7.8	7.909

Table 1: Exact solution for different horizon $t_{\max}$

- $P = \{1, 2 \ldots, 9\}$, in fact points on a small chess-board 3 × 3
- $A = \{1, 2\}$
- $NaturalEvolve(t, e) = \max(x(t-1, e) + 1, 5)$
- C is given by neiborhood on the chess-board, e.g. $(1, 2), (1, 4)$ or $(5, 6) \in C$.

This problem can be solved for small $t_{\max}$ exactly. The graph given by C evidently does not contain Hamiltonian cycle which can be shown by contradiction. Thus, there is no possibility to go through the graph in a cyclical way [12] which would enable efficient patroling [2]. The dynamics, i.e. linear grow up to 5 disables trivial solutions where agent have always a vertex with maximal possible value available for next turn. This example is therefore complex enough for testing. Furthermore, it is demonstrative, so the solution can be insighted easily.

This experiment was tested. For $t_{\max} = 10$, the exact dynamic programming solution required 14 hours and 21 minutes. Optimal values are given in Table 1. Markedly, for these cases the average gain of one time instant grows, as the third line shows.

6.1 Evolutionary solution

Because of high time consumption of the exact solution (growing exponentially), the above proposed evolutionary algorithm was used. It was tested with respect to different parameters and different horizons $t_{\max}$. The parameters were as follows:

- Maximal computer time - how long may the method try to find the solution maximally. Value: $10^{-5}2^{t_{\max}}$ in days.
- Population size - how many solutions are in the population. Tested values: $\{10, 50, 100, 1000\}$
- Copying ratio - how many of best solutions will be copied for the next generation. The absolute number is rounded up (ceiled). Tested values: $\{0.01, 0.1, 0.3\}$
- Sampling ratio - how many of worst solutions will be replaced by random ones. The absolute number is rounded down (floored). Tested values: $\{0, 0.1, 0.3\}$
- Resetting parameter - how many usuccesfull iterations cause the reset (as a ratio of unsuccessfull iterations to all iterations). Used value: 0.5.

Population size	Copying ratio	Sampling ratio
10.0000	0.3000	0.1000
10.0000	0.3000	0.3000
50.0000	0.3000	0.3000

Table 2: Best parameters for $t_{\max} = 7$

$t_{\max}$	systematic	genetic
1	9.03E-07	5.44E-09
2	0.00E+00	1.85E-09
3	1.74E-07	0.00E+00
4	2.35E-06	4.48E-07
5	1.81E-05	4.00E-06
6	1.58E-04	3.52E-05
7	1.20E-03	2.06E-04
8	1.01E-02	1.10E-04
9	7.44E-02	2.79E-04
10	6.48E-01	1.15E-03
11	4.76E+00	3.01E-03

Table 3: Comparing results achieved by evolution and direct solution

All combination of these parameters (in fact of the first three ones) were tested. Each combination and $t_{\max} = 7$ was tested 100 times since the methods are stochastic so the statistical comparison is possible. Comparing results by the t-test, the best results were achieved in three cases shown in Table 2. For these parameters the function of the algorithm was tested up to $t_{\max} = 12$. The results were significantly better as for the direct solution, as shown in Table 3. If we fit the data by an exponential model $f(t_{\max}) = ab^{t_{\max}}$, we obtain $b = 7.28$ for systematic solution, but $b = 2.25$ for the evolutionary solution.

7 Further work

In next steps, I intend to address similar problem for continuous cases, i.e. $P \subset \mathbb{R}^n$. The exact solution will be hardly detected (maybe by variational calculus), but more practical problems can be solved, e.g. wild-fire extinguishing, difussion models etc. The agents will become more deliberated and independent. They will coordinate their actions by sharing information and knowledge. They may consider more objectives (egoistic, altruistic, ecological).

Other, significantly different improvement is to consider the environment as a cognitive map or neural network with switching on and off. But this idea is maybe too challenging.

Next, the algorithm could be improved by storing so far best strategies and substrategies. If there is an evidence a substrategy is optimal, it can be used directly and the alternatives has not to be evaluated. This may require advanced data structures.

Finally, optimal parameters for the optimiztion may be found by an external optimization method for global optimization.

8 Conclusion

One of the most important abilities that is discussed in the multi-agent systems, is the cooperation. In the case of evolutionary and dynamic programming approaches, there was a full coordination since the agents were considered as only one agent with several outputs, in fact. On the other way, some reactive solutions did not reflect other agents completely, e.g. the Random Walk, Greedy Crowd, or Aroma Tracking. However, a kind of indirect coordination was in place by the pheromomone methods and Greedy Lions and Hyenas.

In real system with renewable resources, however the full coordination is not possible and no coordination (or only implicit coordination) is not desirable. This open quite wide space for negotiation situations that are usualy very complex and seems to be complex also in this field.

References

[1] F. Boschetti and M. Brede. *An information-based adaptive strategy for resource exploitation in competitive scenarios.* Technological Forecasting and Social Change **76** (2009), 525 – 532. Evolutionary Methodologies for Analyzing Environmental Innovations and the Implications for Environmental Policy.

[2] Y. Chevaleyre. *Theoretical analysis of the multi-agent patrolling problem.* Intelligent Agent Technology, IEEE / WIC / ACM International Conference on **0** (2004), 302–308.

[3] A. Colorni, M. Dorigo, F. Maffioli, V. Maniezzo, G. Righini, and M. Trubian. *Heuristics from nature for hard combinatorial optimization problems.* International Transactions in Operational Research **3** (1996), 1–21.

[4] R. Damania, R. Damania, R. Damania, E. Barbier, E. Barbier, and E. Barbier. Lobbying, trade and renewable resource harvesting, (2001).

[5] S. Dreyfus. *Richard bellman on the birth of dynamic programming.* Operations Research **50** (January 2002), 48–51.

[6] U. Endriss, N. Maudet, F. Sadri, and F. Toni. *Negotiating socially optimal allocations of resources.* Journal of Artificial Intelligence Research **25** (2006).

[7] M. Lettau and H. Uhlig. *Rules of thumb versus dynamic programming.* AMERICAN ECONOMIC REVIEW **89** (1999), 148–174.

[8] P. Lilienthal, T. Lambert, and P. Gilman. *Computer modeling of renewable power systems.* In 'Encyclopedia of Energy', C. J. Cleveland, (ed.), Elsevier (2004), 633 – 647.

[9] N. V. Long and S. Wang. *Resource-grabbing by status-conscious agents.* Journal of Development Economics **89** (2009), 39 – 50.

[10] K. Macek. Multiagent exploation from renewable resources. In 'Doktorandske dny'. Czech Technical University in Prague, (2008).

[11] K. Macek. Reinforcement learning parameterization: Softmax between exploration and exploation. In 'Proceedings of the 17th International Conference on Process Control '09'. Slovak University of Technology in Bratislava, (2009).

[12] J. Matousek and J. Nesetril. *Invitation to Discrete Mathematics.* Oxford University Press, (1998).

[13] K. Mause. *The tragedy of the commune: Learning from worst-case scenarios.* Journal of Socio-Economics **37** (2008), 308 – 327.

[14] L. Neves, L. Dias, C. Antunes, and A. Martins. *Structuring an mcda model using ssm: A case study in energy efficiency.* European Journal of Operational Research **199** (2009), 834 – 845.

[15] A. Nobre, J. Musango, M. de Wit, and J. Ferreira. *A dynamic ecological-economic modeling approach for aquaculture management.* Ecological Economics **68** (2009), 3007 – 3017.

[16] G. Perez-Verdin, Y.-S. Kim, D. Hospodarsky, and A. Tecle. *Factors driving deforestation in common-pool resources in northern mexico.* Journal of Environmental Management **90** (2009), 331 – 340.

[17] W. M. Spears. Adapting crossover in evolutionary algorithms. In 'Proceedings of the Fourth Annual Conference on Evolutionary Programming', 367–384. MIT Press, (1995).

[18] J.-J. Wang, Y.-Y. Jing, C.-F. Zhang, and J.-H. Zhao. *Review on multi-criteria decision analysis aid in sustainable energy decision-making.* Renewable and Sustainable Energy Reviews **13** (2009), 2263 – 2278.

[19] M. Yagiura and T. Ibaraki. *The use of dynamic programming in genetic algorithms for permutation problems.* European Journal of Operational Research **92** (1996), 387–401.

[20] M. Yagiura and T. Ibaraki. *On metaheuristic algorithms for combinatorial optimization problems.* The Transactions of the Institute of Electronics, Information and Communication Engineers **2** (2001), 83–1.

Quantitative Analysis of Numerical Solution for the Gray-Scott Model

Jan Mach

3rd year of PGS, email: jan.mach@fjfi.cvut.cz
Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, CTU in Prague
advisor: Michal Beneš, Dept. of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague

Abstract. In this contribution we study one particular reaction-diffusion system – the Gray Scott model. We focused on quantitative comparison of two numerical schemes which solve the model in 2D. One is based on FDM, the other is based on FEM and uses the mass-lumping technique. Both schemes are explicit and uses structured numerical grids. Modified Runge-Kutta method with adaptiv time-stepping is used for time-integration. Our numerical simulations suggest that for certain combinations of initial data and model parameter values we may not get an agreement of numerical results provided by these numerical schemes while refinning the numerical grid. Example results are given.

Abstrakt. V příspěvku se zabýváme kvantitativním porovnáím dvou numerických schémat na řešení Grayova-Scottova modelu ve 2D. Jedno je založeno na FDM, druhé na FEM s využitím metody mass-lumping. Obě schémata jsou explicitní a využívají strukturované sítě. K integraci v čase využíváme modifikovanou Runge-Kuttovu metodu s adaptivní volbou časového kroku. Naše numerické simulace ukazují, že existují kombinace počátečních dat a parametru modelu pro které porovnávaná numerická schémata poskytují rozdílné výsledky.

1 Introduction

Reaction-diffusion systems are a class of systems of partial differential equations of parabolic type. It includes mathematical models describing various phenomena e.g. in the fields of physics, biology and chemistry. Gray-Scott model is one of these models. It was first introduced in 1984 by P. Gray and S. K. Scott [1]. It is a mathematical description of the following autocatalytic chemical reaction

$$U + 2V \longrightarrow 3V,\ V \longrightarrow P, \tag{1}$$

where U, V are reactants and P is product of the reaction. Chemical substance U is being continuously added into the reactor and the product P is being continuously removed from the reactor during the reaction. Later it has been extensively studied e.g. by Wei [2], Winter [3], Ueyama [5], Dkhil [6], Doelman [7]. This model is well known to exhibit rich dynamics, see e.g. Nishiura [4]. There exist chemical systems exhibiting features similar to those of the Gray-Scott model, see e.g. [8] and references therein.

2 Problem formulation

We study the Gray-Scott in 2D. Assume that $\Omega \equiv (0,L) \times (0,L)$ is an open square representing the square reactor where the chemical reaction (1) takes place, $\partial\Omega$ is its boundary and ν is its outer normal. Then initial-boundary value problem for the Gray-Scott model we solve is a system of two partial differential equations of parabolic type

$$\begin{aligned} \frac{\partial u}{\partial t} &= a\Delta u - uv^2 + F(1-u), \\ \frac{\partial v}{\partial t} &= b\Delta v + uv^2 - (F+k)v \quad \text{in } \Omega\times(0,T) \end{aligned} \tag{2}$$

with initial conditions $u(\cdot,0) = u_{ini}$, $v(\cdot,0) = v_{ini}$ and zero Neumann boundary conditions $\frac{\partial u}{\partial \nu} \mid_{\partial\Omega}= 0$, $\frac{\partial v}{\partial \nu} \mid_{\partial\Omega}= 0$. The system of PDEs (2) modelling the reaction (1) may be rewriten in several dimensionless forms. We use one which is used also e.g. in [3, 8, 9]. In the system (2) u, v are unknown functions representing concentrations of chemical substances U, V. Parameter F denotes the rate at which the chemical substance U is being added during the chemical reaction, $F + k$ is the rate of $V \to P$ transformation and a, b are constants characterizing the environment where the chemical reaction takes place.

3 Numerical schemes

Computational studies of the Gray-Scott model show difficulties in convergence. We compare two numerical schemes for the initial-boundary value problem defined in Sect. 2 in order to disclose details of these problems. Both of them are based on the method of lines. For spatial discretization we used structured numerical grids consisting of squares for the finite difference method (see Fig. 1a) and of triangles for the finite elements method (see Fig. 1b). To solve resulting systems of ordinary differential equations Runge-Kutta-Merson method (see [11], [12] or [13]) was used.

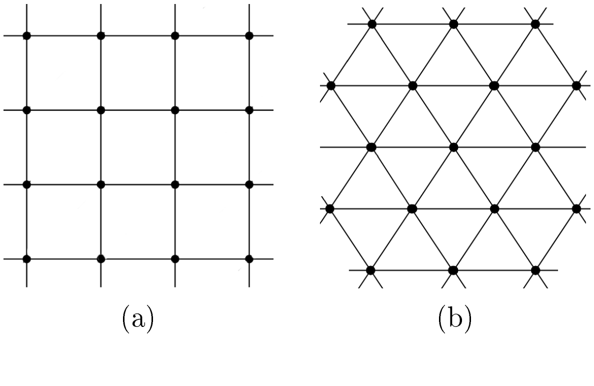


Figure 1: Structured numerical grids for FDM based numerical scheme (a) and for FEM based numerical scheme (b) we used for our numerical simulations.

3.1 FDM based numerical scheme

Let h be mesh size such that $h = \frac{L}{N-1}$ for some $N \in \mathbf{N}^+$. We define numerical grid as a set

$$\begin{aligned} \omega_h &= \{(ih, jh) \mid i = 1, \dots, N-2, j = 1, \dots, N-2\}, \\ \overline{\omega}_h &= \{(ih, jh) \mid i = 0, \dots, N-1, j = 0, \dots, N-1\}. \end{aligned}$$

For function $u : \mathbf{R^2} \to \mathbf{R}$ we define a projection on $\overline{\omega}_h$ as $u_{ij} = u(ih, jh)$. We introduce finite differences

$$\begin{aligned} u_{x_1,ij} &= \frac{u_{i+1,j} - u_{i,j}}{h}, u_{\overline{x}_1,ij} = \frac{u_{i,j} - u_{i-1,j}}{h} \\ u_{x_2,ij} &= \frac{u_{i,j+1} - u_{i,j}}{h}, u_{\overline{x}_2,ij} = \frac{u_{i,j} - u_{i,j-1}}{h}, \end{aligned}$$

and define approximation Δ_h of the Laplace operator Δ as $\Delta_h u_{ij} = u_{\overline{x}_1 x_1,ij} + u_{\overline{x}_2 x_2,ij}$. Then semi-discrete scheme has the following form

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} u_{ij}(t) &= a\Delta_h u_{ij} + F(1 - u_{ij}) - u_{ij} v_{ij}^2, \\ \frac{\mathrm{d}}{\mathrm{d}t} v_{ij}(t) &= b\Delta_h v_{ij} - (F + k) v_{ij} + u_{ij} v_{ij}^2, \end{aligned} \tag{3}$$

plus discrete initial and boundary conditions.

3.2 FEM based numerical scheme

To induce the semi-discrete scheme we begin with variational formulation of the problem in Sect. 2. Let $\varphi_1(x), \varphi_2(x) \in C_0^\infty(\Omega)$ be test functions and denote $f_1(u, v) = F(1 - u) - uv^2$, $f_2(u, v) = -(F + k)v + uv^2$ denote right-hand sides of system (2). Using standard approach (see [10]) we induce weak formulation of the problem

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}(u, \varphi_1) + a(\nabla u, \nabla \varphi_1) &= (f_1, \varphi_1), \\ \frac{\mathrm{d}}{\mathrm{d}t}(v, \varphi_2) + b(\nabla v, \nabla \varphi_2) &= (f_2, \varphi_2), \\ u(\cdot, 0) &= u_{ini}, \\ v(\cdot, 0) &= v_{ini}, \end{aligned} \tag{4}$$

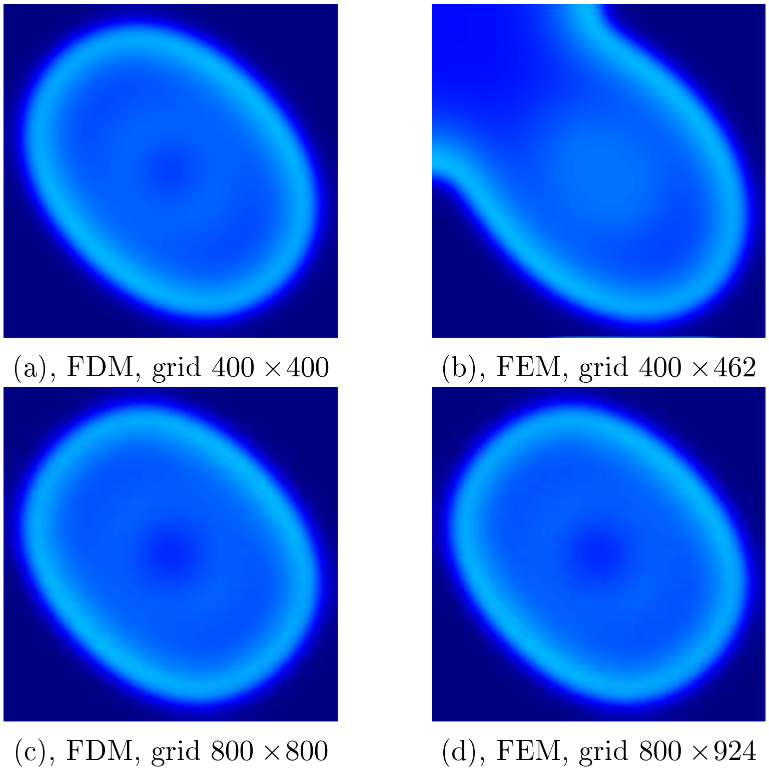
with solution u, v from the Sobolev space $W_2^{(1)}(\Omega)$. We are looking for Galerkin approximation $u_h(t) = \sum_{i=1}^N \alpha_i(t)\Phi_i$, $v_h(t) = \sum_{i=1}^N \beta_i(t)\Phi_i$ of this weak solution in the finite dimensional space $S_h \subset W_2^{(1)}(\Omega)$, where $\Phi_1, \dots, \Phi_N$ are its basis functions. Functions α_i, β_i are real functions which we get using common technique as solutions of initial value problems. Choosing basis functions Φ_i in the form of pyramidal functions $\Phi_i(P_j) = \delta_{ij}$ for all grid nodes P_j, and using mass-lumping we can rewrite the problem for finding functions α_i, β_i in the following form

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}u_{ij}(t) &= \frac{2a}{3h^2}[u_{i+1,j} + u_{i+1,j+1} + u_{i,j-1} + u_{i,j+1} + u_{i-1,j} + \\
&\quad + u_{i-1,j+1} - 6u_{ij}] + F(1-u_{ij}) - u_{ij}v_{ij}^2 \\
\frac{\mathrm{d}}{\mathrm{d}t}v_{ij}(t) &= \frac{2b}{3h^2}[v_{i+1,j} + v_{i+1,j+1} + v_{i,j-1} + v_{i,j+1} + v_{i-1,j} + \\
&\quad + v_{i-1,j+1} - 6v_{ij}] - (F+k)v_{ij} + u_{ij}v_{ij}^2 \qquad (5)
\end{aligned}$$

plus corresponding initial and boundary conditions.

4 Numerical simulations

We performed a series of computations to compare our 2D numerical schemes. According to our results the Gray-Scott model is sensitive on the mesh parameter size, which means, the numerical solution may change notably when refining the computational grid.



(a), FDM, grid 400 ×400

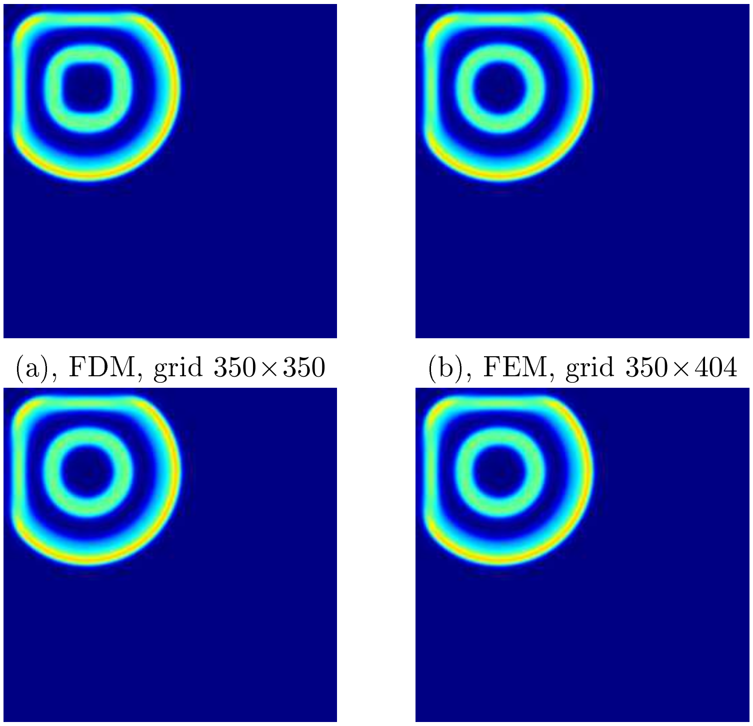
(b), FEM, grid 400 ×462

(c), FDM, grid 800 ×800

(d), FEM, grid 800 ×924

Figure 2: Dependence of pattern in numerical solution on numerical scheme and grid size for given model parameters ($a = 1 \cdot 10^{-5}$, $b = 1 \cdot 10^{-5}$, $F = 0.025$, $k = 0.05$, $L = 0.5$) and initial data (three pulses along minor diagonal) at fixed time $t{=}3000$.

We met initial data and model parameter values combinations for which the following situations occured. First, we have results where FDM based numerical scheme is less



(a), FDM, grid 350×350

(b), FEM, grid 350×404

(c), FDM, grid 800×800

(d), FEM, grid 800×924

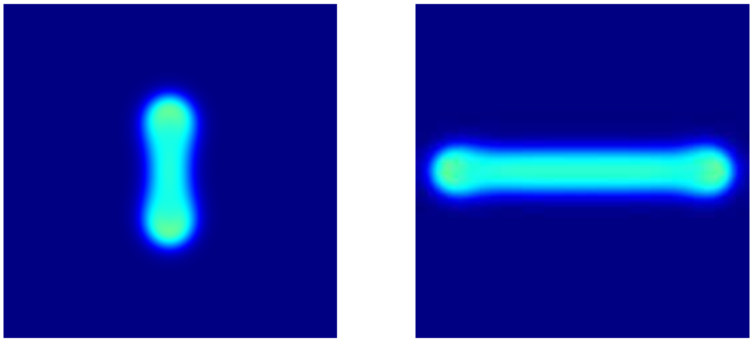
Figure 3: Dependence of pattern in numerical solution on numerical scheme and grid size for given model parameters ($a = 1 \cdot 10^{-5}$, $b = 1 \cdot 10^{-6}$, $F = 0.025$, $k = 0.05$, $L = 0.5$) and initial data (single pulse in upper left corner) at fixed time $t{=}400$.

dependent on the space stepping, that is, the numerical results are visually more similar in wider range of mesh parameter sizes then in case of the FEM based numerical scheme. We have also results, where the FEM based numerical scheme is less dependent on the space stepping. We were able to see agreement in numerical results obtained in both of these cases from certain mesh parameter size. But we have also met combinations where we were not able to see the solutions becoming visually more and more similar while refining the numerical grid. Examples are given below.

In our numerical simulations we use square domain $\Omega \equiv (0.0, 0.5) \times (0.0, 0.5)$. Initial data are considered such that $u_{ini} + v_{ini} = 1$ hold within the computational domain Ω and v_{ini} consists of one or several spots.

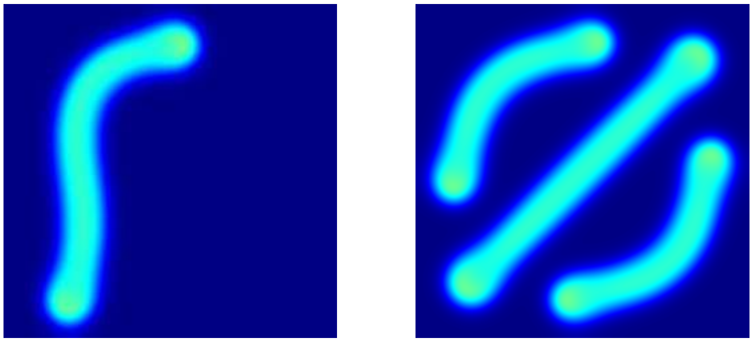
In [7] is suggested in agreement with our experience that when one of the concentrations is large, then the second one is small. That means that seeing the pattern in solution for one concentration we can have a rough idea how the pattern in the second solution component looks like. That is why we show in figures below only spatial distribution of concentration v in domain Ω. Dark blue implies almost zero concentration, lighter color means higher concentration.

In Fig. 2 we demonstrate the case, where FDM based numerical scheme produces solution less dependent on mesh parameter size. We choosed solution at time $t = 3000$. It can be seen that the pattern in solution by the FDM based numerical scheme (left column) do not change notably between selected coarser and finer grid as apposed to



(a), FDM, grid 800×800 (b), FEM, grid 800×924

Figure 4: Dependence of pattern in numerical solution on numerical scheme and grid size for given model parameters ($a = 2 \cdot 10^{-5}$, $b = 1 \cdot 10^{-5}$, $F = 0.0737$, $k = 0.061882$, $L = 0.5$) and initial data (one spot in the middle of the domain) at fixed time $t = 8000$.



(a), grid 800×800 (b), grid 800×800

Figure 5: For the same model parameters as in Fig. 4 an agreement of numerical results was observed for different initial data.

the solutions provided by the second numerical scheme (right column). Here we can see difference in the upper left corner of the domain. From numerical grid of size 800×800 and corresponding size of triangle grid above (mesh parameter $h \approx 6 \cdot 10^{-4}$ and smaller) we were able to see agreement of numerical results in this case.

In Fig. 3 we demonstrate the case, where FEM based numerical scheme produces solution less dependent on mesh parameter size for certain model parameters and initial data combination. We choosed solution at time $t = 400$. It can be seen that solution by the FEM based numerical scheme (right column) do not change notably between coarser and finer grid as apposed to the solutions provided by the second numerical scheme (left column). Here we can see difference in the shape of interior object. In the solution by the FDM based scheme at coarser grid of 350×350 we can see the object to be more square-like. It is changing into perfect circle with finer grid of 800×800. Refining the numerical grid helped to obtain agreement of numerical results.

In Fig. 4 we demonstrate the case, where we were not able to obtain agreement of numerical results by the numerical schemes from Sect. 3. Using the same model parameters and initial data we could see lines growing in ortogonal directions. Depicted are solutions at time $t = 8000$. Each of numerical schemes provided the same pattern in wide range of grid sizes. We tried to succesively refine the numerical grid up to 2000×2000 and corresponding size of triangle grid. We also tried to perform simulations for the same model parameters (see Fig. 4) and different initial data. When starting simulation with single pulse in upper left corner we could see an agreement of numerical results. The solution at time $t = 15000$ is depicted in Fig. 5a. The same situation occured when using three pulse along minor diagonal. Solution at time $t = 4000$ is depicted in Fig. 5(b).

Acknowledgement

The work has been performed under the Project HPC-EUROPA (RII3-CT-2003-506079), with the support of the European Community - Research Infrastructure Action under the FP6 "Structuring the European Research Area" Programme.

Partial support of the project of "Necas Center for Mathematical modeling", No. LC06052 and of the project "Applied Mathematics in Physics and Technical Sciences", No. MSN6840770010 of the Ministry of Education, Youth and Sports of the Czech Republic is acknowledged.

References

[1] Gray, P., Scott, S.K. *Autocatalytic reactions in the isothermal, continuous stirred tank reactor: oscillations and instabilities in the system $A + 2B \rightarrow 3B$, $B \rightarrow C$.* In Chem. Eng. Sci. *39*, pp. 1087-1097 (1984).

[2] Wei, J. *Pattern formation in two-dimensional Gray-Scott model: existence of single-spot solutions and their stability.* In Physica D *148*, pp. 20-48. (2001).

[3] Wei, J., Winter, M. *Asymmetric spotty patterns for the Gray-Scott model in* $\mathbf{R}^2$. In Stud. Appl. Math. *110*, no. 1, pp. 63-102. (2003).

[4] Nishiura Y., Ueyama D. *Self-Replication, Self-Destruction, and Spation-Temporal Chaos in the Gray-Scott model.* In Forma *15*, pp. 281-289, (2000).

[5] Nishiura Y., Ueyama D. *Spatio-temporal chaos for the Gray-Scott model.* In Physica D, vol. *150*, pp. 137-162 (2001).

[6] Dkhil F., Logak E., Nishiura Y. *Some analytical results on the Gray-Scott model.* In Asymptotic Analysis, vol. *39*, pp. 225-261, IOS Press (2004).

[7] Doelman A. et al. *Pattern formation in the one-dimensional Gray-Scott model.* In Nonlinearity *10*, pp. 523-563, (1997).

[8] Mazin, W., Rasmussen K.E., Mosekilde E., et al. *Pettern formation in the bistable Gray-Scott model.* In: Mathematics na Computers in Simulations *40*, pp. 371-396. Elsevier Science (1996).

[9] McGough J.S., Riley K. *Pattern Formation in the Gray-Scott model.* In Nonlinear Analysis: Real World Application *5*, pp. 105-121. Elsevier Science (2004).

[10] Thomée, V. *Galerkin Finite Element Methods for Parabolic Problems. Springer-Verlag Berlin Heidelberg.* (1997).

[11] Holodniok, M., Klíč, A., Kubíček, M., Marek, M. In *Methods for analysis of non-linear dynamical models (Metody analýzy nelineárních dynamických modelu).* Academia Praha (1986), in Czech.

[12] Šembera, J. and Beneš, M. *Nonlinear Galerkin Method for Reaction-Diffusion Systems Admitting Invariant Regions.* In Journal of Computational and Applied Mathematics, Vol 136/1-2, pp 163-176.

[13] Minárik V., Kratochvíl J., Mikula K. and Beneš M. *Numerical simulation of dislocation dynamics.* In Numerical Mathematics and Advanced Applications, ENUMATH 2003 (peer reviewed proceedings), pp. 631–641, eds. Feistauer M., Dolejší V., Knobloch P., Najzar K., Springer Verlag, (2004).

Dicrete Dislocation Dynamics*

Petr Pauš

3rd year of PGS, email: pauspetr@fjfi.cvut.cz
Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, CTU in Prague
advisor: Michal Beneš, Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University

Abstract. This paper deals with the numerical simulation of dislocation dynamics. Dislocations are described by means of the evolution of a family of closed and open smooth curves $\Gamma(t) : S^1 \to \mathbb{R}^2$, $t \geqq 0$. The curves are driven by the normal velocity v which is the function of curvature κ and the position vector $x \in \Gamma(t)$. In this case the equation is defined this way: $v = -\kappa + F$. The equation is solved using direct approach by two numerical schemes, ie. semi-implicit and semi-discrete. Results of the dislocation dynamics simulation are presented.

Abstrakt. Tento článek se zabývá numerickou simulací dislokační dynamiky. Dislokace jsou popsány pomocí časového vývoje množiny uzavřených a otevřených hladkých křivek $\Gamma(t) : S^1 \to \mathbb{R}^2$, $t \geqq 0$. Vývoj křivek je ovlivňován normálovou rychlostí v, jenž je funkcí křivosti κ a polohového vektoru $x \in \Gamma(t)$. V tomto případě má rovnice tvar $v = -\kappa + F$. Rovnice je řešena přímou metodou pomocí dvou různých numerických schémat, semi-implicitním a semi-diskrétním. Výsledky simulace dislokační dynamiky jsou také uvedeny.

1 Introduction

Tthe dislocations are defined as irregularities or errors in crystal structure of the material. The presence of dislocations strongly influences many of material properties. Plastic deformation in crystalline solids is carried by dislocations. Theoretical description of dislocations is widely provided in literature such as [17–19]. Dislocation is a line defect of the crystalline lattice. Along the dislocation curve the regularity of the crystallographic arrangement of atoms is disturbed. The dislocation can be represented by a curve closed inside the crystal or by a curve ending on surface of the crystal. At low homologous temperatures the dislocations can move only along crystallographic planes (gliding planes) with the highest density of atoms. The motion results in mutual slipping of neighboring parts of the crystal along the gliding planes.

This justifies the importance of developing suitable mathematical models [9–14]. From the mathematical point of view, the dislocations are defined as smooth closed or open plane curves which evolve in time. Their motion is two-dimensional. The evolving curves can be mathematically described in several ways. One possibility is to use the *level-set method* [1–3], where the curve is defined by the zero level of some surface function. One can also use the *phase-field method* [4]. Finally, it is possible to use the *direct (parametric) method* [6, 7] where the curve is parametrized in the usual way.

[*] This work is supported by grant no. MSM 6840770010, project no. LC06052 of Nečas center for mathematical modeling.

2 Parametric description

When using the parametric approach, the planar curve $\Gamma(t)$ is described by a smooth time-dependent vector function

$$X : S \times I \rightarrow \mathbb{R}^2,$$

where $S = \langle 0, 1\rangle$ is a fixed interval for the curve parameter and $I = \langle 0, T\rangle$ is the time interval. The curve $\Gamma(t)$ is then given as the set

$$\Gamma(t) = \{X(u,t) = (X^1(u,t), X^2(u,t)), u \in S\}.$$

The family of curves satisfies the equation of motion

$$v = -\kappa + F, \tag{1}$$

where v is the normal velocity of the curve evolution, κ is the curvature, and F is the forcing term which can depend on position vector x and time t.

The evolution law (1) is transformed into the parametric form. The unit tangential vector $\vec{T}$ is defined as $\vec{T} = \partial_u X/|\partial_u X|$. The unit normal vector $\vec{N}$ is perpendicular to the tangential vector and $\vec{N} \cdot \vec{T} = 0$ holds. In case of closed curve, $\vec{N}$ is the outer vector to the interior of the curve. In case of open curve, $\vec{N}$ has a selected, pre-defined direction (e.g., upwards). The orientation of the curve is clockwise. The curvature κ is expressed as

$$-\kappa = \frac{\partial_u X^\perp}{|\partial_u X|} \cdot \frac{\partial_{uu} X}{|\partial_u X|^2} = \vec{N} \cdot \frac{\partial_{uu} X}{|\partial_u X|^2},$$

where $X^\perp$ is a vector perpendicular to X. The normal velocity v is defined as the time derivative of X projected into the normal direction,

$$v = \partial_t X \cdot \frac{\partial_u X^\perp}{|\partial_u X|}.$$

The equation (1) can now be written as

$$\partial_t X \cdot \frac{\partial_u X^\perp}{|\partial_u X|} = \frac{\partial_{uu} X}{|\partial_u X|^2} \cdot \frac{\partial_u X^\perp}{|\partial_u X|} + F(X,t),$$

which holds provided

$$\partial_t X = \frac{\partial_{uu} X}{|\partial_u X|^2} + F(X,t)\frac{\partial_u X^\perp}{|\partial_u X|}. \tag{2}$$

This equation is accompanied by the periodic boundary conditions for closed curves, or by fixed-end boundary condition for open curves, and by the initial condition. These conditions are considered similarly as in [6]. The solution of (2) exhibits a natural redistribution property which is useful for short-time curve evolution [8,12]. The redistribution of curve discretization points is operated by tangential forces discussed below.

The term $\partial_{uu}X/|\partial_u X|^2$ in (2) contains a tangential component which makes the curve points to move along the curve. To modify or cancel this tangential force, a term α in the tangential direction can be considered as follows

$$\partial_t X = \frac{\partial_{uu}X}{|\partial_u X|^2} - \alpha\frac{\partial_u X}{|\partial_u X|} + F(X,t)\frac{\partial_u X^\perp}{|\partial_u X|}. \tag{3}$$

Hence the tangential term contained in equation (3) has the form

$$\alpha = \frac{\partial_{uu}X \cdot \partial_u X}{|\partial_u X|^3}. \tag{4}$$

Then the equation without a tangential force has the following form:

$$\partial_t X = \frac{\partial_{uu}X}{|\partial_u X|^2} - \frac{\partial_{uu}X \cdot \partial_u X}{|\partial_u X|^4}\partial_u X + F(X,t)\frac{\partial_u X^\perp}{|\partial_u X|}. \tag{5}$$

This equation is not suitable for numerical simulations because the curve points do not move along the curve and can accumulate in some parts or move from each other in other parts of the curve. This can cause a slow-down in computation. The equation (2) is better for numerical simulations but still for long time simulations similar accumulation of points can happen. Additional algorithm for tangential redistribution of points has to be considered.

For long time computations with time and space variable external force $F(X,t)$, the algorithm for curvature adjusted tangential velocity is used. This algorithm moves points along the curve according to the curvature, i.e., areas with higher curvature contain more points than areas with lower curvature. This improves numerical stability and also accuracy of computation. The term α is based on the relative local length between points. Details are described in [16]. Another approach based on finite-element discretization of equations for curve parametrization is in [21], where existing multiple junctions are treated as well.

3 Numerical scheme

For numerical approximation we consider a regularized form of (3) which reads as

$$\partial_t X = \frac{\partial_{uu}X}{Q(\partial_u X)^2} - \alpha\frac{\partial_u X}{Q(\partial_u X)} + F(X,t)\frac{\partial_u X^\perp}{Q(\partial_u X)}, \tag{6}$$

where $Q(x_1,x_2) = \sqrt{x_1^2 + x_2^2 + \varepsilon^2}$ with ε being a small parameter. Two numerical schemes are used for numerical solution of the differential equation (3), i.e., backward Euler semi-implicit and semi-discrete method of lines. With two numerical schemes it is possible to compare the solution and error of computation.

In the semi-discrete scheme of method of lines, spatial derivatives are approximated by fourth-order central differences. The first derivative is approximated as

$$\partial_u X|_{u=jh} \approx \left[\frac{X^1_{j-2} - 8X^1_{j-1} + 8X^1_{j+1} - X^1_{j+2}}{12h}, \frac{X^2_{j-2} - 8X^2_{j-1} + 8X^2_{j+1} - X^2_{j+2}}{12h}\right],$$

and the second one as

$$\partial_{uu}X|_{u=jh} \approx \Big[\frac{-X^1_{j-2}+16X^1_{j-1}-30X^1_j+16X^1_{j+1}-X^1_{j+2}}{12h^2}, \frac{-X^2_{j-2}+16X^2_{j-1}-30X^2_j+16X^2_{j+1}-X^2_{j+2}}{12h^2}\Big],$$

where X^i_j denotes an approximation of $X^i(jh,\cdot), i \in \{1,2\}, h = 1/m$. Here m is a number of intervals dividing S. The difference expressions above are denoted as X_u for the first difference and X_{uu} for the second difference.

The equation (6) in the semi-discrete scheme of method of lines has the following form:

$$\frac{dX_j}{dt} = \frac{X_{uu,j}}{Q^2(X_{u,j})} - \alpha_j \frac{X_{u,j}}{Q(X_{u,j})} + F(X_j,t)\frac{X^{\perp}_{u,j}}{Q(X_{u,j})}, \qquad j = 1,\cdots,m-1, t \in (0,T), \tag{7}$$

where again $Q(x_1,x_2) = \sqrt{x_1^2+x_2^2+\varepsilon^2}$, $X^{\perp}_{u,j}$ is a vector perpendicular to $X_{u,j}$, and α_j is the redistribution coeficient. The term with ε serves as a regularization to avoid singularities when the curvature tends to infinity. This scheme is solved by the fourth order Runge-Kutta method [5].

The second approach uses the backward Euler semi-implicit scheme. In this case lower order differences are used. The first derivative is discretized by the backward difference as follows

$$\partial_u X|_{u=jh} \approx \left[\frac{X^1_j - X^1_{j-1}}{h}, \frac{X^2_j - X^2_{j-1}}{h}\right],$$

and the second derivative as

$$\partial_{uu}X|_{u=jh} \approx \left[\frac{X^1_{j+1}-2X^1_j+X^1_{j-1}}{h^2}, \frac{X^2_{j+1}-2X^2_j+X^2_{j-1}}{h^2}\right].$$

The approximation of the first derivative is denoted as $X_{\bar{u},j}$ and of the second derivative as $X_{\bar{u}u,j}$.

The semi-implicit scheme for equation (3) has the following form

$$X^{k+1}_j - \tau\frac{X^{k+1}_{\bar{u}u,j}}{Q^2(X^k_{\bar{u},j})} + \tau\alpha_j\frac{X^{k+1}_{\bar{u},j}}{Q(X^k_{\bar{u},j})} = X^k_j + \tau F(X^k_j, k\tau)\frac{X^{\perp k}_{\bar{u},j}}{Q(X^k_{\bar{u},j})}, \qquad j = 1,\cdots,m-1, k = 0,\cdots,N_T-1, \tag{8}$$

where $Q(x_1,x_2)$, $X^{\perp}_{\bar{u},j}$, m, and α_j have the same meaning as for semi-discrete scheme. $X^k_j \approx X(jh,k\tau)$, τ is a time step and N_T is the number of time steps. The matrix of the system (8) for each component of X^{k+1} has the following tridiagonal structure:

$$\begin{pmatrix} 1+\frac{2\tau}{h^2Q^2}-\frac{\tau\alpha}{hQ} & \frac{-\tau}{h^2Q^2} & 0 & \cdots \\ \frac{-\tau}{h^2Q^2}+\frac{\tau\alpha}{hQ} & \ddots & \ddots & \ddots \\ 0 & \ddots & & \\ \vdots & \ddots & & \end{pmatrix}.$$

The scheme (8) is solved for each k by means of matrix factorization. Since there are two components of X, two linear systems are solved in each timestep.

4 Topological changes

In curve dynamics in general, and in dislocation dynamics in particular, topological changes may occur (e.g., connecting or splitting, closing of open curves, etc.). The parametric approach does not handle them intrinsically, and we therefore need an additional algorithm allowing for such changes of discretized curves.

The algorithm we present is not supposed to be universal for every situation and possibility. Main purpose is to simulate topological changes that can happen during dislocation dynamics (see [13]), i.e., topological changes such as merging or splitting of curves, closing of open curves, etc. As the initial condition, we consider only curves which do not intersect itself and do not touch each other. The orientation of curves is clockwise. The algorithm is designed for topological changes of curves which touch only at one point. More complex changes can be treated by multiple application of the algorithm in one timestep. The evolution after merging or splitting behaves as expected. Normal vectors and evolution speed correspond to the situation captured by the level-set method. The results of the algorithm were compared with the level-set method in [15].

Let us consider two closed or open curve parametrizations discretized as $X = \{x_1, x_2, \cdots, x_n\}$ and $Y = \{y_1, y_2, \cdots, y_m\}$ in $\mathbb{R}^2$. Curves evolve independently according to the equation (3). The algorithm for merging two curves is as follows:

1. Compute the distance between X and Y and find one point from each curve where the minimum is reached. Let us denote the distance as d, the point from X as x_{max} and from Y as y_{max}.

2. Check if the distance d between curves is smaller than a given tolerance δ. If not, compute new timestep and go to 1.

3. Create new empty curve Z. We must take into account the type of merged curves. Merging two closed curves will produce one closed curve. Merging one open and one closed curve will produce one open curve and merging two open curves will produce two open curves.

4. Copy points from X from the begining (i.e., from x_1) up to x_{max} to Z.

5. Copy points from Y from y_{max} up to the end (i.e., up to y_m) to Z.

6. Copy points from Y from the begining (i.e., from y_1) up to y_{max} to Z.

7. Copy points from X from x_{max} up to the end (i.e., up to x_n) to Z.

8. Delete X and Y.

9. Compute a new timestep for Z and go to 1.

We also consider that one curve can intersect itself and thus split itself into 2 parts. Let us consider a closed or open curve discretized as $X = \{x_1, x_2, \cdots, x_n\}$. The curve evolves independently according to the equation (3). The algorithm for splitting into two curves is as follows:

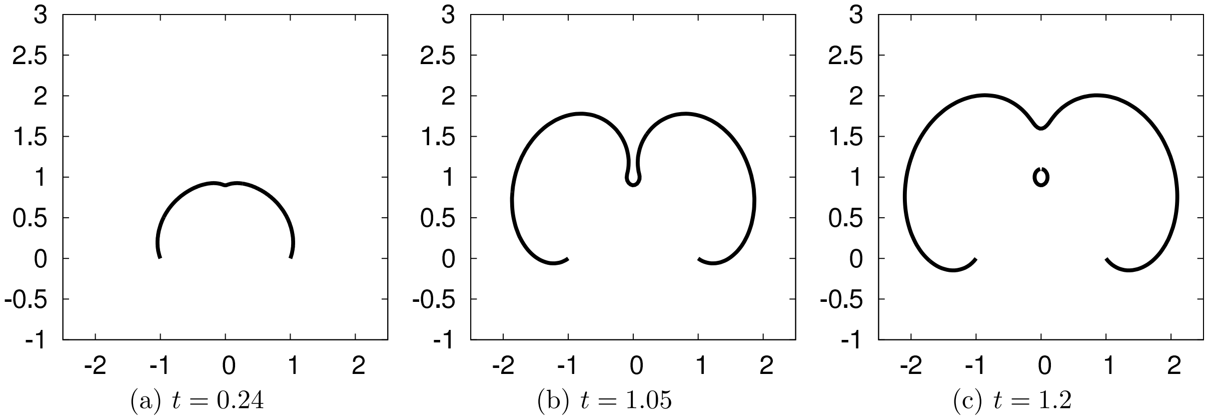
1. Compute the distance between points in X and find two points where the minimum was reached. Let us denote the distance as d, and the points as x_{max1} and x_{max2}. We do not consider several points in the neighbourhood of each point when measuring the distance to avoid finding minimal distance for two neighbor points. The number has to be computed according to the value of a given tolerance δ (see the next step). We recommend to omit all points with the distance smaller than at least 4δ.
2. Check if the distance d between points is smaller than a given tolerance δ. If not, compute new timestep and go to 1.
3. Create two new empty curves X_{new1} and X_{new2}. If X is an open curve, X_{new1} will be open and X_{new2} closed curve. If X is a closed curve then X_{new1} and X_{new2} will be closed curves.
4. Copy points from X from the begining (i.e., from x_1) up to x_{max1} to X_{new1}.
5. Copy points from X from x_{max1} up to x_{max2} to X_{new2}.
6. Copy points from X from x_{max2} up to the end (i.e., up to x_n) to X_{new1}.
7. Delete X.
8. Compute new timestep for X_{new1} and X_{new2} and go to 1.

The numerical simulation is shown in Figure 4.

5 Application in dislocation dynamics

Dislocation curves as defects in material evolve in time. The dislocation evolution history contains shape changes of open curves, closing of open dislocation curves up to collision of dipolar loops (see [17]). Interaction of dislocation curves and dipolar loops has been studied, e.g., in [9–11].

Dislocations can interact with other defects through the stress field. In this case, dislocation curve can be blocked by a potential barrier. Fig. 1 illustrates the evolution of a dislocation curve through an obstacle in material. In the example, the obstacle has a form of circle located at $[0, 1]$ with a radius of 0.1. Due to external force, the dislocation curve expands but the obstacle blocks the evolution. The curve surrounds it. At a certain time, it touches itself and splits into two curves, an open curve and a closed curve. Closed curve cannot evolve anymore because of the obstacle. Open curve continues expansion. The simulation was performed with the following parameters. The number of discretization points is $M = 200$, the external force applied to the dislocation $F_D = -5.0$, the force of the obstacle $F_O = 20.0$, the time of simulation $t \in (0, 1.2)$. The initial condition was given as a half-circle with a radius of 0.5 located at $[0, 0]$.



(a) $t = 0.24$ (b) $t = 1.05$ (c) $t = 1.2$

Figure 1: Evolution through a strong obstacle, $F_O = 20.0$, $F_D = -5.0$, $t \in (0, 1.2)$, curve discretized by $M = 200$ nodes.

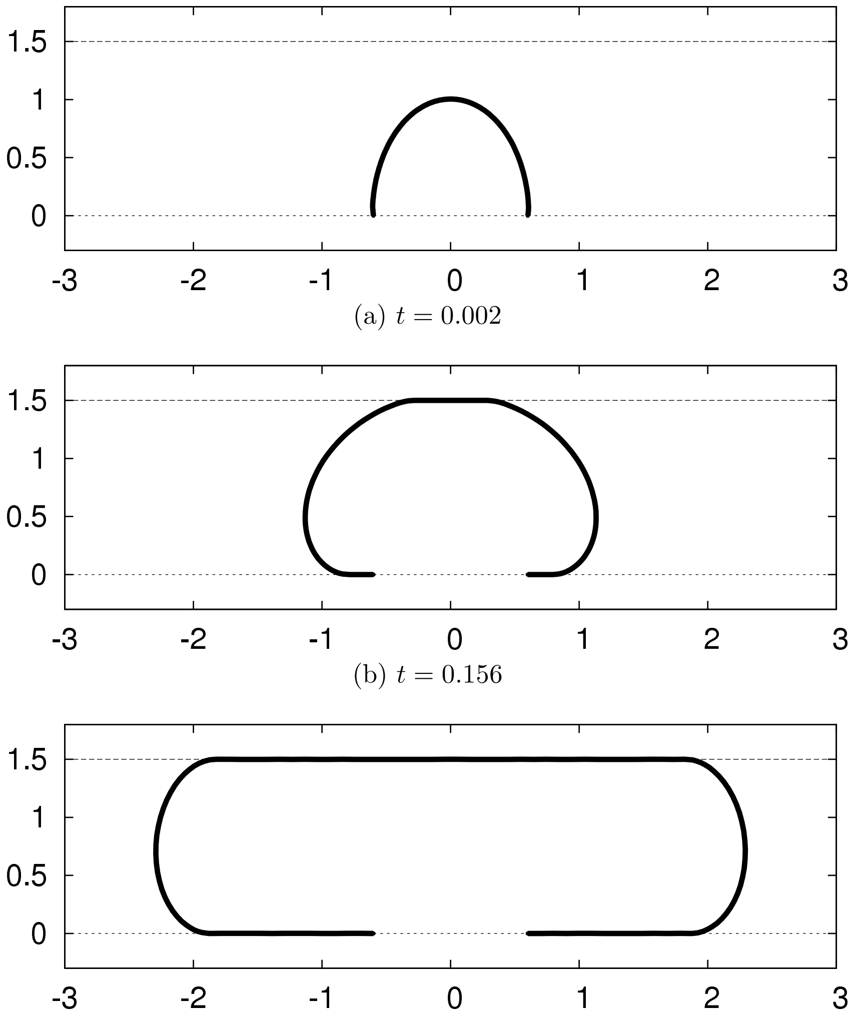
Fig. 2 illustrates the behavior of an open dislocation curve in an infinite channel. The channel is created by a spatially variable external force $F_C = 20.0$ for $y < 0.0$ and $y > 1.5$. The curve expands upwards due to external force $F_D = -5.0$ applied to the dislocation. The upper channel wall restricts its movement. The curve can therefore evolve aside only. The algorithm for curvature adjusted redistribution of points allows to rarify number of discretization points along straight parts of the dislocation and accumulate discretization points at parts with higher curvature. This results into more accurate and faster computations. The parameters of simulation are $t \in (0, 0.444)$, $M = 128$.

The simulation of cross-slip of two dislocations is shown in Fig. 5. The dislocations are moving in the channel created by a spatially variable external force $F_C = 20.0$ for $y < -0.6$ and $y > 0.6$. At a certain time, they touch each other and connect. In real material, each dislocation can evolve in a different parallel plane. This case is not yet covered by the described model. Parameters of the simulation are $F_D = -5.0$, $t \in (0, 0.164)$, $M = 75$.

The example in Fig. 4 shows the simulation of the Frank-Read mechanism (see [17, 19])which describes how new dislocation loops are created. An external force $F_D = -2.5$ is applied to the dislocation line forcing the curve to expand until it touches itself. At this moment, the curve splits into two parts, i.e., dipolar loop and dislocation line. The loop continues in expansion. The dislocation line will again undergo the same process. The initial condition was given as a half-circle with a radius of 1.0 located at $[0, 0]$. Parameters of the simulation are $t \in (0, 2.9)$, $M = 200$.

6 Conclusion

The simulation of dislocation dynamics is important in practice as dislocations affect many material properties. Dislocation dynamics can be mathematically simulated by the mean curvature flow. We presented a method based on a parametric approach and two numerical schemes. We applied the model to situations similar to the real context including a mechanism of creating new dislocations (i.e., Frank-Read source). The scheme had to be improved by an algorithm for tangential redistribution of points and by an



(a) $t = 0.002$

(b) $t = 0.156$

(c) $t = 0.444$

Figure 2: Single dislocation in an infinite channel, $F_C = 20.0$, $F_D = -5.0$, $t \in (0, 0.444)$, curve discretized by $M = 128$ nodes.

algorithm for topological changes for parametric model.

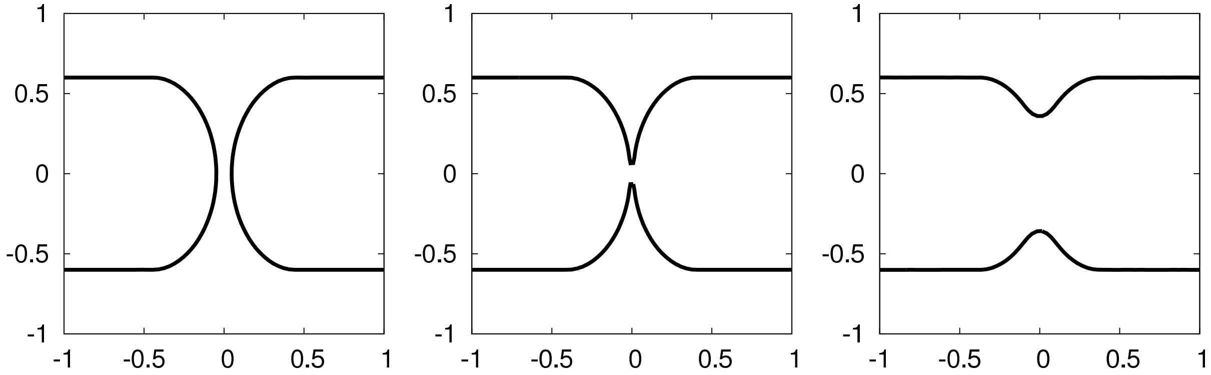
References

[1] S. Osher, R. P. Fedkiw: Level set methods and dynamic implicit surfaces. Springer, New York 2003

[2] J. A. Sethian: Level set methods and fast marching methods. Cambridge University Press, Cambridge 1999.

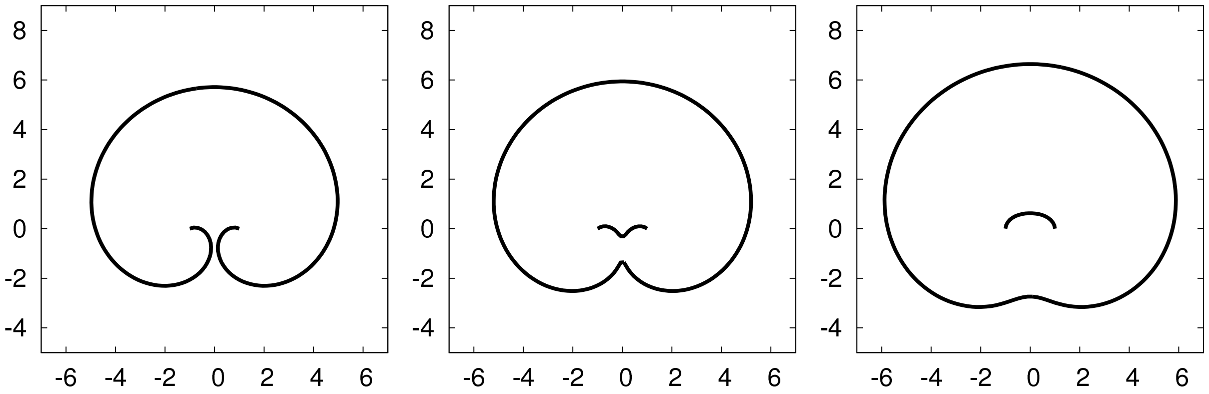
[3] G. Dziuk, A. Schmidt, A. Brillard, and C. Bandle: Course on mean curvature flow. Manuscript 75p., Freiburg, 1994.

[4] M. Beneš: Phase field model of microstructure growth in solidification of pure substances. Acta Math. Univ. Comenian 70 (2001), 123–151.



(a) $t = 0.128$ (b) $t = 0.138$ (c) $t = 0.146$

Figure 3: Merging two dislocations in a channel, $F_C = 20.0$, $F_D = -5.0$, $t \in (0, 0.146)$, curve discretized by $M = 75$ nodes.



(a) $t = 2.5$ (b) $t = 2.6$ (c) $t = 2.9$

Figure 4: Frank-Read source, $F_D = -2.5$, $t \in (0, 2.9)$, curve discretized by $M = 200$ nodes.

[5] M. Beneš: Mathematical analysis of phase-field equations with numerically efficient coupling terms. Interfaces and Free Boundaries 3 (2001), 201–221.

[6] K. Deckelnick, G. Dziuk: Mean curvature flow and related topics. Frontiers in numerical analysis (2002), 63–108.

[7] K. Mikula, D. Ševčovič: Computational and qualitative aspects of evolution of curves driven by curvature and external force. Computing and Visualization in Science Vol. 6 No. 4 (2004), 211–225.

[8] K. Mikula, D. Ševčovič: Evolution of plane curves driven by a nonlinear function of curvature and anisotropy. SIAM Journal on Applied Mathematics Vol. 61 No. 5 (2001), 1473–1501.

[9] V. Minárik, J. Kratochvíl: Dislocation Dynamics – Analytical Description of the Interaction Force between Dipolar Loops. Kybernetika 43 (2007), 841–854

[10] V. Minárik, J. Kratochvíl, K. Mikula: Numerical Simulation of Dislocation Dynamics by Means of Parametric Approach. In: Proceedings of the Czech Japanese Seminar in Applied Mathematics (M. Beneš, J. Mikyška, T. Oberhuber, eds), Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, Prague 2005, pp. 128–138.

[11] V. Minárik, J. Kratochvíl, K. Mikula, M. Beneš: Numerical simulation of dislocation dynamics. In: Numerical Mathematics and Advanced Applications – ENUMATH 2003 (M. Feistauer, V. Dolejší, P. Knobloch, K. Najzar, eds), Springer Verlag, New York 2004, pp. 631–641.

[12] P. Pauš: Numerical simulation of dislocation dynamics. In: Proceedings of Slovak-Austrian Congress, Magia (M. Vajsáblová, P. Struk, eds.), Bratislava, pp. 45–52.

[13] P. Pauš, M. Beneš: Topological changes for parametric mean curvature flow. In: Proceedings of Algoritmy conference, Podbanské, 2009.

[14] P. Pauš, M. Beneš: Direct approach to mean-curvature flow with topological changes, to appear in Kybernetika (2009)

[15] P. Pauš, M. Beneš: Comparison of methods for mean curvature flow. (in preparation).

[16] D. Ševčovič, S. Yazaki: On a motion of plane curves with a curvature adjusted tangential velocity. In: http://www.iam.fmph.uniba.sk/institute/sevcovic/papers/cl39.pdf, arXiv:0711.2568, 2007.

[17] T. Mura: Micromechanics of Defects in Solids. Springer, 1987.

[18] F. Kroupa: Long-range elastic field of semi-infinite dislocation dipole and of dislocation jog. Phys. Status Solidi 9 (1965), 27–32.

[19] J.P. Hirth, J. Lothe: Theory of Dislocations. John Willey, New York, 1982.

[20] S. Altschuler, M.A. Grayson: Shortening space curves and flow through singularities. J. Differential Geom. 35 (1992), 283–298.

[21] J.W. Barrett, H. Garcke, R. Nürnberg: On the variational approximation of combined second and fourth order geometric evolution equations. SIAM J. Sci. Comp. 29 (2007), 1006–1041.

Palindromes in Infinite Words[*]

Štěpán Starosta

2nd year of PGS, email: sstarosta@seznam.cz
Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, CTU in Prague
advisor: Edita Pelantová, Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, CTU in Prague

Abstract. We study infinite words $\mathbf{u}$ which generalize Sturmian words for a multiliteral alphabet $\mathcal{A}$. The generalization is defined by property

$$\mathcal{PE}: \quad \text{every palindrome of } \mathbf{u} \text{ has exactly one palindromic extension in } \mathbf{u}\,.$$

We focus on infinite words with the language closed under reversal and complexity $(\#\mathcal{A}-1)n+1$. A sufficient and necessary condition of property $\mathcal{PE}$ is shown.

Abstrakt. Studujeme nekonečná slova $\mathbf{u}$, která zobecňují Sturmovská slova na vícepísmenné abecedě $\mathcal{A}$. Zobecnění je definováno vlastností

$$\mathcal{PE}: \quad \text{každý palindrom } \mathbf{u} \text{ má právě jedno palindromické rozšíření v } \mathbf{u}\,.$$

Věnujeme se nekonečným slovům s jazykem uzavřeným na reverzi a komplexitou $(\#\mathcal{A}-1)n+1$. Dokážeme postačující a nutnou podmínku vlastnosti $\mathcal{PE}$.

1 Introduction

Combinatorics on words is a relatively new research domain. The first works date to the beginning of the 20th century. Since then the domain has greatly developed and found many applications in different fields. These fields are for instance theoretical informatics, symbolic dynamics and music theory. One of the most intensively studied topic are Sturmian words which appeared already in 1940. They were introduced by Morse and Hedlund [8] as aperiodic words with the minimal possible complexity, i.e., with the complexity $\mathcal{C}(n) = n + 1$ for any $n \in \mathbb{N}$. ($\mathbb{N}$ stands for nonnegative integers.) The complexity of a given infinite word $\mathbf{u}$ is the function $\mathcal{C} : \mathbb{N} \mapsto \mathbb{N}$ defined by

$$\mathcal{C}(n) = \text{number of factors of length } n \text{ occurring in } \mathbf{u}\,,$$

where a factor stands for a subword of finite length that can be read in the $\mathbf{u}$ without skipping any letters. The set of all factors occurring in $\mathbf{u}$ is called the *language* of $\mathbf{u}$ and denoted throughout this paper by $\mathcal{L}(\mathbf{u})$. There exist many equivalent definitions of Sturmian words. Already in [8], Sturmian words are characterized by their balance property. In the center of our attention will be another characterization of Sturmian

[*] This work has been suported by the Czech Science Foundation grant 201/09/0584 and by the grant LC06002 of the Ministry of Education, Youth, and Sports of the Czech Republic.

words, proved in [5]. This characterization uses the palindromic complexity of $\mathbf{u}$, which is the function $\mathcal{P} : \mathbb{N} \mapsto \mathbb{N}$ defined by

$$\mathcal{P}(n) = \text{number of palindromic factors of length } n \text{ occurring in } \mathbf{u}\,.$$

A palindrome is a word read the same backward and forward. Droubay and Pirillo proved that an infinite word $\mathbf{u}$ is Sturmian if and only if its palindromic complexity is

$$\mathcal{P}(n) = \begin{cases} 1 & \text{if } n \text{ is even,} \\ 2 & \text{if } n \text{ is odd.} \end{cases}$$

Since the empty word is the only palindrome of length 0 and the letters of the alphabet $\mathcal{A}$ are the only palindromes of length 1 in $\mathbf{u}$, the previous property can be rewritten in a compact form for infinite words over a 2-letter alphabet as

$$\mathcal{P}(n) + \mathcal{P}(n+1) = 3 \quad \text{for any } n \in \mathbb{N}.$$

Being inspired by Sturmian words, we generalize the previous property for infinite words over any alphabet $\mathcal{A}$ as

$$\mathcal{P}: \qquad \mathcal{P}(n) + \mathcal{P}(n+1) = 1 + \#\mathcal{A} \quad \text{for any } n \in \mathbb{N}.$$

It is again readily seen that the property $\mathcal{P}$ is equivalent to the property

$$\mathcal{P}(n) = \begin{cases} 1 & \text{if } n \text{ is even,} \\ \#\mathcal{A} & \text{if } n \text{ is odd.} \end{cases}$$

Examples of infinite words over multiliteral alphabets satisfying the property $\mathcal{P}$ are Arnoux-Rauzy words (also called strict episturmian words, see [7]) and nondegenerate words coding the r-interval exchange transformation with the permutation $\pi = (r, r-1, r-2, \ldots, 2, 1)$ (see [1]).

When studying in details the proof of Droubay and Pirillo, we learn that a binary word $\mathbf{u}$ is Sturmian if and only if $\mathbf{u}$ satisfies the following condition

$$\mathcal{PE}: \qquad \text{any palindromic factor of } \mathbf{u} \text{ has a unique palindromic extension in } \mathbf{u}\,.$$

In other words, for any palindrome $p \in \mathcal{L}(\mathbf{u})$ there exists a unique letter $a \in \mathcal{A}$ such that $apa \in \mathcal{L}(\mathbf{u})$. In fact, our two examples of words with the property $\mathcal{P}$ - namely Arnoux-Rauzy words (see [7]) and words coding interval exchange - have even the property $\mathcal{PE}$ (see [1]).

Infinite words over a multiliteral alphabet satisfying the property $\mathcal{P}$ or $\mathcal{PE}$ may be understood as one of the possible generalizations of Sturmian words. It is evident that $\mathcal{PE}$ implies $\mathcal{P}$. The validity of $\mathcal{P}$ or $\mathcal{PE}$ guarantees that the language $\mathcal{L}(\mathbf{u})$ contains infinitely many distinct palindromic factors. Such a language need not contain the mirror image of every its element, i.e., be closed under reversal. Nevertheless in the sequel, we concentrate on the study of words whose language is closed under reversal. It is readily seen that such words are recurrent and their Rauzy graphs have a non-trivial automorphism that will serve as a powerful tool in our consideration.

For the description of $\mathcal{PE}$ we will use a notion characterizing how many different extensions of a given factor w exist in $\mathcal{L}(\mathbf{u})$. The number denoted $b(w)$ called the bilateral order of factor w is defined by

$$b(w) := \#\{xwy \in \mathcal{L}(\mathbf{u}) \mid x, y \in \mathcal{A}\} - \#\{xw \in \mathcal{L}(\mathbf{u}) \mid x \in \mathcal{A}\} - \#\{wy \in \mathcal{L}(\mathbf{u}) \mid y \in \mathcal{A}\} + 1.$$

Factors having their bilateral order 0 are called ordinary. We will prove the following theorem:

Theorem 1. *Let* $\mathbf{u}$ *be an infinite word over a k-letter alphabet whose set of factors* $\mathcal{L}(\mathbf{u})$ *is closed under reversal and whose factor complexity is* $\mathcal{C}(n) = (k-1)n + 1$. *Then* $\mathcal{L}(\mathbf{u})$ *satisfies* $\mathcal{PE}$ *if and only if all factors from* $\mathcal{L}(\mathbf{u})$ *are ordinary.*

The property $\mathcal{P}$ is more difficult to characterize. However we know (see [2]) a sufficient condition for a ternary alphabet.

Theorem 2 ([2])**.** *An infinite ternary word whose language is closed under reversal has the property* $\mathcal{P}$ *if its complexity satisfies* $\mathcal{C}(n) = 2n + 1$.

Let's recall another result from [2]. Properties $\mathcal{P}$ and $\mathcal{PE}$ are in fact equivalent on a binary alphabet. However, they are no longer equivalent on a ternary alphabet as there exists a word on a ternary alphabet that serves as a counterexample.

It is interesting to mention three corollaries of the previous theorems. Vuillon [9] showed that a binary infinite word is Sturmian if and only if each of its factors has exactly two return words, i.e., Sturmian words are precisely binary words satisfying the property

$$\mathcal{R}: \quad \text{any factor of } \mathbf{u} \text{ has exactly } \#\mathcal{A} \text{ return words.}$$

In the paper [3], it is shown that a uniformly recurrent (every factor occurs with bounded gaps) word with all factors having nonnegative bilateral order has the property $\mathcal{R}$ if and only if its complexity satisfies $\mathcal{C}(n) = (\#\mathcal{A} - 1)n + 1$. Using Theorem 1 we conclude as follows.

Corollary 3. *Let* $\mathbf{u}$ *be a uniformly recurrent infinite word over k-letter alphabet whose language is closed under reversal and satisfies* $\mathcal{C}(n) = (k-1)n + 1$. *Then* $\mathcal{PE}$ *implies* $\mathcal{R}$.

In the same paper [3], it is also shown that a ternary infinite uniformly recurrent word $\mathbf{u}$ has the property $\mathcal{R}$ if and only if its complexity satisfies $\mathcal{C}(n) = 2n + 1$ and $\mathbf{u}$ has no maximal left special factor. As the last two conditions imply that all factors are ordinary, this gives rise to the following corollary.

Corollary 4. *For ternary infinite words with the language closed under reversal,* $\mathcal{R}$ *implies* $\mathcal{PE}$.

Theorem 2 says that for infinite words whose language is closed under reversal and whose complexity satisfies $\mathcal{C}(n) = 2n + 1$, the following equation holds

$$\mathcal{P}(n) + \mathcal{P}(n+1) = 2 + \mathcal{C}(n+1) - \mathcal{C}(n)\,. \tag{1}$$

Infinite words fulfilling the above equation are in a certain sense the richest in palindromes, since according to [1], any infinite word whose language is closed under reversal satisfies

$$\mathcal{P}(n) + \mathcal{P}(n+1) \leq 2 + \mathcal{C}(n+1) - \mathcal{C}(n) \,.$$

(In fact, the above relation is stated in [1] only for uniformly recurrent words, however the proof requires only recurrent words.)

In [6], it is shown that for infinite words with the language closed under reversal, the words defined by the equation (1) are exactly the so-called rich words. Let us recall that an infinite word is called rich if every its prefix w contains $|w| + 1$ distinct palindromes. Consequently, we have the following corollary.

Corollary 5. *Infinite ternary words with the language closed under reversal and the complexity $\mathcal{C}(n) = 2n + 1$ are rich.*

In Section 2, we recall basic notions from combinatorics on words. Section 3 contains the proof of Theorem 1.

2 Preliminaries

By $\mathcal{A}$ we denote a finite set of symbols, usually called *letters*; the set $\mathcal{A}$ is therefore called an *alphabet*. A finite string $w = w_0 w_1 \ldots w_{n-1}$ of letters of $\mathcal{A}$ is said to be a *finite word*, its length is denoted by $|w| = n$. Finite words over $\mathcal{A}$ together with the operation of concatenation and the empty word ε as the neutral element form a free monoid $\mathcal{A}^*$. The map

$$w = w_0 w_1 \ldots w_{n-1} \quad \mapsto \quad \overline{w} = w_{n-1} w_{n-2} \ldots w_0$$

is a bijection on $\mathcal{A}^*$, the word $\overline{w}$ is called the *reversal* or the *mirror image* of w. A word w which coincides with its mirror image is a *palindrome*.

Under an *infinite word* $\mathbf{u}$ we understand an infinite string $\mathbf{u} = u_0 u_1 u_2 \ldots$ of letters from $\mathcal{A}$. A finite word w is a *factor* of a word v (finite or infinite) if there exist words $w^{(1)}$ and $w^{(2)}$ such that $v = w^{(1)} w w^{(2)}$. If $w^{(1)} = \varepsilon$, then w is said to be a *prefix* of v, if $w^{(2)} = \varepsilon$, then w is a *suffix* of v. The *language* $\mathcal{L}(\mathbf{u})$ of an infinite word $\mathbf{u}$ is the set of all its factors. The factors of $\mathbf{u}$ of length n form the set denoted by $\mathcal{L}_n(\mathbf{u})$. Using this notation, we may write $\mathcal{L}(\mathbf{u}) = \cup_{n \in \mathbb{N}} \mathcal{L}_n(\mathbf{u})$. We say that the language $\mathcal{L}(\mathbf{u})$ is *closed under reversal* if $\mathcal{L}(\mathbf{u})$ contains with every factor w also its reversal $\overline{w}$.

For any factor $w \in \mathcal{L}(\mathbf{u})$, there exists an index i such that w is a prefix of the infinite word $u_i u_{i+1} u_{i+2} \ldots$. Such an index i is called an *occurrence* of w in $\mathbf{u}$. If each factor of $\mathbf{u}$ has at least two occurrences in $\mathbf{u}$, the infinite word $\mathbf{u}$ is said to be *recurrent*. It is easy to see that if the language of $\mathbf{u}$ is closed under reversal, then $\mathbf{u}$ is recurrent.

A *complete return word* of a factor w is a word $v \in \mathcal{L}(\mathbf{u})$ which has exactly two distinct occurrences of w, one as a prefix and one as a suffix.

The *complexity* of an infinite word $\mathbf{u}$ is a map $\mathcal{C} : \mathbb{N} \mapsto \mathbb{N}$, defined by $\mathcal{C}(n) = \#\mathcal{L}_n(\mathbf{u})$. To determine the increment of the complexity, one has to count the possible extensions of factors of length n. A *right extension* of $w \in \mathcal{L}(\mathbf{u})$ is any letter $a \in \mathcal{A}$ such that $wa \in \mathcal{L}(\mathbf{u})$. The set of all right extensions of a factor w will be denoted by $\mathrm{Rext}(w)$. Of course, any factor of $\mathbf{u}$ has at least one right extension. A factor w is called *right special* if w has at least two right extensions. Clearly, any suffix of a right special factor is right special as well. A right special factor w which is not a suffix of any longer right special factor is called a *maximal right special* factor. Similarly, one can define a *left extension*, a *left special* factor and $\mathrm{Lext}(w)$. We will deal only with recurrent infinite words $\mathbf{u}$. In this case, any factor of $\mathbf{u}$ has at least one left extension. If $a \in \mathcal{A}$ and p is a palindrome and $apa \in \mathcal{L}(\mathbf{u})$, then apa is said to be a *palindromic extension* of p. We say that w is a *bispecial* factor if it is right and left special. The role of bispecial factors for the computation of the complexity can be nicely illustrated on Rauzy graphs.

Let $\mathbf{u}$ be an infinite word and $n \in \mathbb{N}$. The *Rauzy graph* Γ_n of $\mathbf{u}$ is a directed graph whose set of vertices is $\mathcal{L}_n(\mathbf{u})$ and set of edges is $\mathcal{L}_{n+1}(\mathbf{u})$. An edge $e \in \mathcal{L}_{n+1}(\mathbf{u})$ starts at the vertex x and ends at the vertex y if x is a prefix and y is a suffix of e.

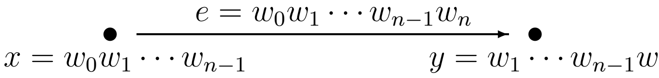

$e = w_0w_1\cdots w_{n-1}w_n$
$x = w_0w_1\cdots w_{n-1}$
$y = w_1\cdots w_{n-1}w_n$

Figure 1: Incidence relation between an edge and vertices in a Rauzy graph.

If the word $\mathbf{u}$ is recurrent, the graph Γ_n is strongly connected for every $n \in \mathbb{N}$, i.e., there exists a directed path from every vertex x to every vertex y of the graph.

The reversal mapping acts on a Rauzy graph by sending vertices and edges to their mirror images and changing the orientation of edges.

The *outdegree* (*indegree*) of a vertex $x \in \mathcal{L}_n(\mathbf{u})$ is the number of edges which start (end) in x. Obviously the outdegree of x is equal to $\#\mathrm{Rext}(x)$ and the indegree of x is $\#\mathrm{Lext}(x)$.

A *simple path* is a path in Γ_n which begins with a special factor, ends with a special factor and contains no other special factor. The *reduced Rauzy graph* of order n, denoted Γ'_n, is constructed from Γ_n by considering only special factors, i.e., vertices with indegree or outdegre grater than 1. There is an edge in Γ'_n going from a vertex v to a vertex w if there is a simple path in Γ_n beginning with v and ending with w.

The sum of outdegrees over all vertices is equal to the number of edges in every directed graph. Similarly, it holds for indegrees. In particular, for the Rauzy graph we have

$$\sum_{x\in\mathcal{L}_n(\mathbf{u})} \#\mathrm{Rext}(x) \ = \ \mathcal{C}(n+1) \ = \sum_{x\in\mathcal{L}_n(\mathbf{u})} \#\mathrm{Lext}(x)\,.$$

The first difference of complexity $\Delta\mathcal{C}(n) = \mathcal{C}(n+1) - \mathcal{C}(n)$ is thus given by

$$\Delta\mathcal{C}(n) = \sum_{x\in\mathcal{L}_n(\mathbf{u})} \big(\#\mathrm{Rext}(x) - 1\big) \ = \sum_{x\in\mathcal{L}_n(\mathbf{u})} \big(\#\mathrm{Lext}(x) - 1\big)\,.$$

Let us restrict our consideration to recurrent words, then a non-zero contribution to $\Delta\mathcal{C}(n)$ is given only by those factors $x \in \mathcal{L}_n(\mathbf{u})$, for which $\#\mathrm{Rext}(x) \geq 2$ or $\#\mathrm{Lext}(x) \geq 2$, i.e., for right or left special factors. The last relation can be rewritten as

$$\Delta\mathcal{C}(n) = \sum_{x\in\mathcal{L}_n(\mathbf{u}),\ x \text{ right special}} \big(\#\mathrm{Rext}(x) - 1\big) \ = \sum_{x\in\mathcal{L}_n(\mathbf{u}),\ x \text{ left special}} \big(\#\mathrm{Lext}(x) - 1\big) \ .$$

Cassaigne [4] introduced the following formula for the second difference of complexity $\Delta^2\mathcal{C}(n)$. Since every factor of length $n+2$ can be written as xwy, where $x, y \in \mathcal{A}$ and $w \in \mathcal{L}(\mathbf{u})$, it holds

$$\mathcal{C}(n+2) = \sum_{w\in\mathcal{L}_n(\mathbf{u})} \#\{xwy \mid xwy \in \mathcal{L}(\mathbf{u})\}.$$

We will denote the set of both-sided extensions of w by $\mathrm{Bext}(w)$:

$$\mathrm{Bext}(w) := \{xwy \mid xwy \in \mathcal{L}(\mathbf{u}), x, y \in \mathcal{A}\}.$$

Similarly,

$$\mathcal{C}(n+1) = \sum_{w\in\mathcal{L}_n(\mathbf{u})} \#\mathrm{Lext}(w) = \sum_{w\in\mathcal{L}_n(\mathbf{u})} \#\mathrm{Rext}(w).$$

The second difference of complexity $\Delta^2\mathcal{C}(n) = \Delta\mathcal{C}(n+1) - \Delta\mathcal{C}(n) = \mathcal{C}(n+2) - 2\mathcal{C}(n+1) + \mathcal{C}(n)$ may be obtained as follows

$$\Delta^2\mathcal{C}(n) = \sum_{w\in\mathcal{L}_n(\mathbf{u})} \Big(\#\mathrm{Bext}(w) - \#\mathrm{Lext}(w) - \#\mathrm{Rext}(w) + 1\Big). \tag{2}$$

Denote by $b(w)$ the quantity

$$b(w) := \#\mathrm{Bext}(w) - \#\mathrm{Lext}(w) - \#\mathrm{Rext}(w) + 1.$$

The number $b(w)$ is called the *bilateral order* of the factor w. It is readily seen that if w is not a bispecial factor, then $b(w) = 0$. Bispecial factors will be distinguished according to their bilateral order in the following way

- if $b(w) > 0$, then we call w a *strong* bispecial factor,
- if $b(w) < 0$, then we call w a *weak* bispecial factor,
- if $b(w) = 0$ and w is bispecial, then we call it *ordinary.*

Evidently, for the value of $\Delta^2\mathcal{C}(n)$, only strong and weak bispecial factors are of importance.

3 Proof of Theorem 1

The proof is divided into following lemmas. First we prove the if part of the theorem.

Lemma 6. *Let* $\mathbf{u}$ *be an infinite word whose language is closed under reversal. If a bispecial factor* $w \in \mathcal{L}(\mathbf{u})$ *has its bilateral order* $b(w)$ *even, then its number of palindromic extensions is odd.*

Proof. Let w be a bispecial factor of $\mathcal{L}(\mathbf{u})$ and its bilateral order $b(w)$ is even. As the language is closed under reversal, we have $\#\mathrm{Lext}(w) = \#\mathrm{Rext}(w)$. From the definition of bilateral order one can see that $\#\mathrm{Bext}(w)$ has to be odd. Let $x, y \in \mathcal{A}$. As $xwy \in \mathcal{L}(\mathbf{u})$ implies $ywx \in \mathcal{L}(\mathbf{u})$, it is clear that the number of palindromic extensions xwx has different parity then $b(w)$ and therefore is odd. □

Lemma 7. *Let* $\mathbf{u}$ *be an infinite word whose language is closed under reversal and its factor complexity is* $\mathcal{C}(n) = (\#\mathcal{A} - 1)n + 1$. *If all palindromic factors of* $\mathcal{L}(\mathbf{u})$ *have even bilateral order, then* $\mathcal{PE}$ *is satisfied.*

Proof. It is clear that the property $\mathcal{PE}$ can only be violated on a palindromic bispecial factor.

We suppose that all palindromic factors have even bilateral order. According to the previous lemma they have odd number of palindromic extensions.

Since the language is closed under reversal we have for all n

$$\mathcal{P}(n) + \mathcal{P}(n+1) \leq 2 + \mathcal{C}(n+1) - \mathcal{C}(n). \tag{3}$$

Let w denote the shortest palindromic bispecial factor that doesn't have exactly one palindromic extension. Denote $N = |w|$. Then we have for all $n < N$,

$$\mathcal{P}(n) + \mathcal{P}(n+1) = 2 + \mathcal{C}(n+1) - \mathcal{C}(n),$$

i.e., the maximum number of palindromes is attained.

Since w has to have at least 3 palindromic extensions, oen can see that $\mathcal{P}(N+2) \geq \mathcal{P}(N) + 2$. Thus, for $n = N+1$ we have a contradiction with (3) as the right-hand side is constantly equal to $1 + \#\mathcal{A}$ and the inequality is no longer satisfied. We conclude that $\mathcal{PE}$ holds. □

The last lemma proves the if part of the proof. The following three lemmas will serve to prove the other direction of the equivalence.

Lemma 8. *Let* $\mathbf{u}$ *be a rich infinite word whose set of factors is closed under reversal. Then all its bispecial factors are palindromes.*

Proof. Let us denote by $RS(n)$, $LS(n)$ and $BS(n)$ the number of right special, left special and bispecial factors in $\mathcal{L}_n(\mathbf{u})$.

Let's fix n and consider the reduced Rauzy graph Γ'_n. We have

$$\begin{aligned}\#\{\text{edges in } \Gamma'_n\} &= \sum_{\substack{w\in\mathcal{L}_n(\mathbf{u})\\ \#\mathrm{Rext}(w)>1}} \#\mathrm{Rext}(w) + \sum_{\substack{w\in\mathcal{L}_n(\mathbf{u})\\ \#\mathrm{Lext}(w)>1,\#\mathrm{Rext}(w)=1}} 1 \\ &= \underbrace{\sum_{\substack{w\in\mathcal{L}_n(\mathbf{u})\\ \#\mathrm{Rext}(w)>1}} (\#\mathrm{Rext}(w) - 1)}_{\Delta C(n)} + \underbrace{\sum_{\substack{w\in\mathcal{L}_n(\mathbf{u})\\ \#\mathrm{Rext}(w)>1}} 1}_{RS(n)} + \underbrace{\sum_{\substack{w\in\mathcal{L}_n(\mathbf{u})\\ \#\mathrm{Lext}(w)>1,\#\mathrm{Rext}(w)=1}} 1}_{LS(n)-BS(n)}.\end{aligned}$$

On the other hand we have

$$\begin{aligned}\#\{\text{edges in } \Gamma'_n\} = &\underbrace{\#\{\text{edges in } \Gamma'_n \text{ invariant under reversal}\}}_{P(n)+P(n+1)-\#\{\text{special palindromes of length } n\}} \\ &+ \#\{\text{edges in } \Gamma'_n \text{ not invariant under reversal}\}.\end{aligned}$$

Putting both sides into one equation we find

$$\Delta C(n) + RS(n) + LS(n) - BS(n) = P(n) + P(n+1)$$
$$-\#\{\text{special palindromes of length } n\} + \#\{\text{edges in } \tilde{\Gamma}_n \text{ not invariant under reversal}\}.$$

The last term can be estimated by the lowest number of edges that need to go to a different vertex so that the graph is strongly connected. As $LS(n) = RS(n)$ we have

$$\#\{\text{edges in } \Gamma'_n \text{ not invariant under reversal}\} \geq 2\left(LS(n) - 1\right).$$

After simplification we obtain

$$-BS(n) + \#\{\text{special palindromes of length } n\} \geq P(n) + P(n+1) - \Delta C(n) - 2.$$

Since we assumed $\mathbf{u}$ rich, the right-hand side is equal to 0. Therefore we have

$$BS(n) \leq \#\{\text{special palindromes of length } n\}.$$

As the language is closed under reversal, special palindromes are bispecial factors. The last inequality is in fact an equality, i.e., every bispecial factor is a palindrome. □

Lemma 9. *Let $\mathbf{u}$ be an infinite rich word whose set of factors closed under reversal. Then for a bispecial factor $w \in L(\mathbf{u})$ we have*

$$\#\mathrm{Bext}(w) \geq 2\left(\#\mathrm{Rext}(w) - 1\right).$$

Proof. As w is bispecial, according to the previous lemma it is also a palindrome. Let n denote its length. Let l denote the number of its right (or equally left) extensions $l := \#\mathrm{Rext}(w)$. Let $x_1, \ldots, x_l$ denote the l distinct letters that extend w.

We will look at the Rauzy graph Γ_{n+1} where the extensions of w are vertices. Figure **??** shows all possible edges $x_i w x_j$ connecting these vertices. To evaluate the minimal value for $\#\mathrm{Bext}(w)$, we need to know how many of these edges need to exist in Γ_{n+1}. The minimum value is given by the fact that a Rauzy graph (of a recurrent word) is strongly connected.

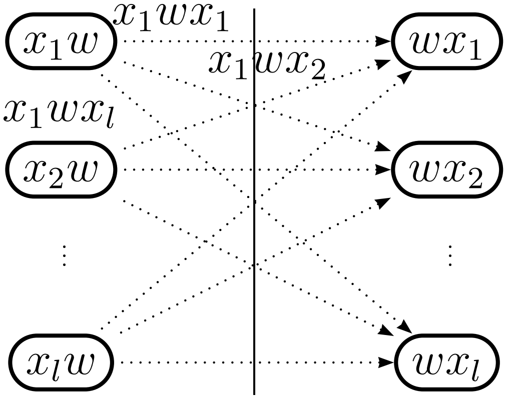


Figure 2: Part of Γ_{n+1}: the vertices containing w are drawn together with all possible edges connecting them. Only edges going from $x_1 w$ are labeled.

Let $1 \geq i, j \geq l$ be two indices. As the word $\mathbf{u}$ is rich, all complete return word to palindromes are palindromes (see [6]). The existence of such return word implies that there is at least one path from wx_i to x_iw in Γ_{n+1}. Let us divide the set $\{x_1, \dots, x_l\}$ into two non-empty disjoint sets X and Y. Let Xw denote the vertices $\{xw \mid x \in X\}$ and wY the set $\{wy \mid y \in Y\}$. Assume there is no edge going from Xw to wY. As Γ_{n+1} is strongly connected, there has to be a path from wX to Yw which doesn't include any other vertex from X or Y than at the beginning and at the end. This implies existence of a word beginning with wx, $x \in X$, and ending with yw, $y \in \mathcal{A}$, without any other occurrence of w. Such word would be a non-palindromic return word of w which is not possible. Together with the independence of choice of the sets X and Y, we can conclude that there is no path going from wx_i to x_jw, $i \neq j$, without passing any other vertex containing w.

Therefore the minimum number of edges x_iwx_j that need to be placed is the minimum number of edges so that the graph is strongly connected while taking into consideration the previous conclusion. As the language is closed under reversal, one can see we need to place at least $l - 1$ pairs x_iwx_j and x_jwx_i which proves the claim. □

Corollary 10. *Let u be an infinite rich word whose set of factors is closed under reversal. Then all its factors have their bilateral order $b(w)$ greater or equal to -1. Furthermore, if for a factor w we have $b(w) = -1$, then w has no palindromic extension.*

Proof. If a factor w is not bispecial, $b(w) = 0$. If w is bispecial, we apply the previous lemma to evaluate its bilateral order:

$$b(w) = \#\mathrm{Bext}(w) - 2\#\mathrm{Rext}(w) + 1 \geq 2\left(\#\mathrm{Rext}(w) - 1\right) - 2\#\mathrm{Rext}(w) + 1 = -1.$$

If the bilateral order is equal to its minimum, $b(w) = -1$, we can see from the proof of the previous lemma that there are no edges x_iwx_i, i.e. w has no palindromic extension. □

Lemma 11. *Let $\mathbf{u}$ be an infinite word over a k-letter alphabet whose set of factors is closed under reversal and complexity satisfies $\mathcal{C}(n) = (k-1)n + 1$. If the property $\mathcal{PE}$ is satisfied, we have $\forall w \in \mathcal{L}(\mathbf{u})$, $b(w) = 0$.*

Proof. The property $\mathcal{PE}$ and complexity $(k-1)n + 1$ imply

$$\mathcal{P}(n) + \mathcal{P}(n+1) = \Delta\mathcal{C}(n) + 2, \quad \text{for all } n.$$

Therefore the word $\mathbf{u}$ is rich.

Let w be a bispecial factor. As it has exactly one palindromic extension, using Corollary 10 we see that we need to add 1 to the minimal value of $b(w)$. Therefore we have $b(w) \geq 0$. On the other hand

$$\sum_{w \in \mathcal{L}_n(\mathbf{u})} b(w) = \Delta^2\mathcal{C}(n) = 0, \quad \text{for all } n.$$

Since all non-bispecial factors have their bilateral order 0, these two conditions imply $b(w) = 0$ for all bispecial factors.

□

The last lemma completes the proof:

Proof of Theorem 1. The only if part is given exactly by the Lemma 11. The if part follows from Lemma 7. □

References

[1] P. Baláži, Z. Masáková, E. Pelantová, *Factor versus palindromic complexity of uniformly recurrent infinite words.* Theoret. Comput. Sci. **380** (2007) 266-275.

[2] L'. Balková, E. Pelantová, Š. Starosta, *Palindromes in infinite ternary words*, RAIRO-Theor. Inf. Appl. PREPRINT (2009).

[3] L'. Balková, E. Pelantová, W. Steiner, *Sequences with Constant Number of Return Words.* Monatsh. Math., **155(3-4)** (2008) 251-263.

[4] J. Cassaigne, *Complexity and special factors.* Bull. Belg. Math. Soc. Simon Stevin 4 **1** (1997) 67-88.

[5] X. Droubay, G. Pirillo, *Palindromes and Sturmian words.* Theoret. Comput. Sci. **223** (1999) 73-85.

[6] M. Bucci, A. De Luca, A. Glen, L. Q. Zamboni, *A connection between palindromic and factor complexity using return words.* Adv. in Appl. Math **42** (2009) 60-74.

[7] J. Justin, G. Pirillo, *Episturmian words and episturmian morphisms.* Theoret. Comput. Sci. **276** (2002), 281-313.

[8] M. Morse, G. A. Hedlund, *Symbolic dynamics II - Sturmian trajectories.* Amer. J. Math. **62** (1940), 1-42.

[9] L. Vuillon, *A characterization of Sturmian words by return words.* Eur. J. Comb. **22** (2001) 263-275.

MEGIDDO: Design and Study of the Diffusion-based MR-DTI Visualization Algorithm

Pavel Strachota

2nd year of PGS, email: pavel.strachota@fjfi.cvut.cz
Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, CTU in Prague
advisor: Michal Beneš, Dept. of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague

Abstract. This contribution describes the proposed neural tract visualization technique based on the MR-DTI data. The cornerstone of the algorithm is a texture diffusion procedure modeled mathematically by the problem for the Allen-Cahn equation with diffusion anisotropy controlled by a tensor field. Focus is put on the issues of the numerical solution of the given problem, using the finite volume method for spatial domain discretization. Several numerical schemes are compared with the aim of reducing the artificial (numerical) isotropic diffusion. The remaining steps of the algorithm are commented on as well, including the acquisition of the tensor field before the actual computation begins and the postprocessing used to obtain the final images. Finally, the visualization results are presented.

Abstrakt. Příspěvek popisuje techniku zobrazování nervových traktů na základě dat získaných metodou MR-DTI. Algortimus je založen na difuzi šumové textury modelované pomocí úlohy pro Allenovu-Cahnovu rovnici s anizotropií difuzního členu řízenou tenzorovým polem. Práce se soustředí na numerické řešení úlohy a metodu konečných objemů použitou při její prostorové diskretizaci. Je srovnáváno několik schémat s cílem omezit numerickou izotropní difuzi. Probrány jsou však i ostatní kroky celé procedury. Nakonec jsou prezentovány výsledky vizualizací.

1 Introduction

The MR-DTI (*Magnetic Resonance Diffusion Tensor Imaging*, [4]) technique belongs to the family of noninvasive medical examination methods based on magnetic resonance phenomenon and indicated in a wide variety of human health problems. In particular, DTI is dedicated to examining anisotropic structures in tissues, such as heart muscle fibers or neural tracts in human brain.

Denote by Ω_0 the examined volume of the brain. By means of DTI, the strength and directional distribution of water molecule diffusion in each volume element (voxel) of Ω_0 is measured and encoded into a symmetric positive definite *diffusion tensor field* $\boldsymbol{D} : \Omega_0 \mapsto \mathbb{R}^{3\times 3}$. At a given point $\boldsymbol{x} \in \Omega_0$, it can be interpreted by the *diffusion ellipsoid* defined as

$$\Gamma\left(\boldsymbol{x}\right) = \left\{ \boldsymbol{\eta} \in \mathbb{R}^3 \middle| \, \boldsymbol{\eta}^T \boldsymbol{D}\left(\boldsymbol{x}\right)^{-1} \boldsymbol{\eta} = 1 \right\}.$$

The diffusion strength along the vector $\boldsymbol{v}$ is proportional to the distance from the origin to $\Gamma\left(\boldsymbol{x}\right)$ in the direction of $\boldsymbol{v}$ and the eigenvalues of $\boldsymbol{D}\left(\boldsymbol{x}\right)$ represent the lengths of the principal axes of $\Gamma\left(\boldsymbol{x}\right)$.

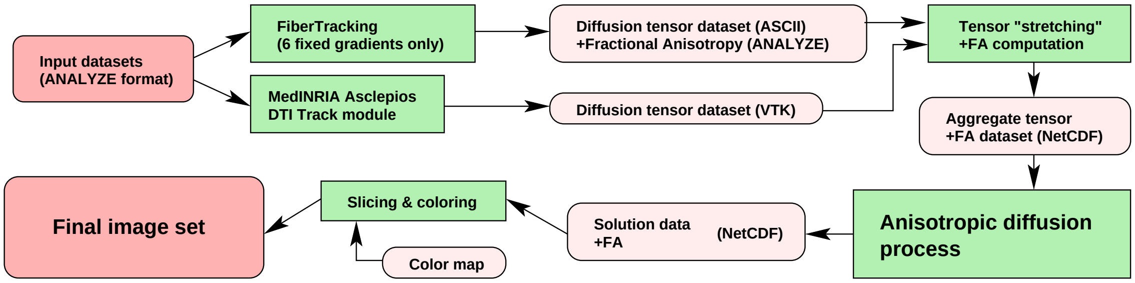


Figure 2.1: MEGIDDO data processing workflow.

It has been observed that diffusion prevails in the direction parallel to the neural fiber tracts [4]. Hence, it is possible to perform MR *tractography* [9], i.e. to recover a model of the fiber bundles by following the pathways of the strongest diffusion. However, some explicit fiber reconstruction algorithms [6, 16] may experience local ambiguities e.g. at fiber crossings as the tensor representation only provides a second order approximation of the general directional distribution.

2 Overview of the proposed visualization algorithm

Principle of the method. We have elaborated an algorithm based on globally imitating the diffusion processes taking place in the brain tissue, similar to the technique introduced in [15]. The idea is to apply an anisotropic diffusion process [12] to a noisy 3D texture contained in a *region of interest* (ROI) $\Omega \subset \Omega_0$. The anisotropy of the diffusion is controlled by the tensor field $\boldsymbol{D}$ so that the distribution of the texture diffusion strength corresponds to the physical process measured in the brain. As a result, the initial noisy image is smeared in such a way that the streamlines of the tensor field become distinguishable. The 3D volume Ω can then be sliced to produce human readable planar images.

The described steps have been implemented in the MEGIDDO (Medical Employment of Generating Images by Degenerate Diffusion Operator) software kit. In the following paragraphs, we will focus on the details of its data processing workflow (see Figure 2.1).

Data acquisition. Raw DWI (*Diffusion Weighted Image*, [4]) datasets are delivered from the scanner either in a proprietary format used for the vendor supplied software or in the well documented ANALYZE 7.5 or DICOM formats. Currently, the DTI module of MedINRIA (developed within the ASCLEPIOS project at INRIA, Sophia Antipolis, France) is used to process these images and to compute the diffusion tensor field, which may also involve thresholding and smoothing to cope with noise in the input data [14].

Fractional anisotropy and diffusion ellipsoid stretching. Denote by $\lambda_1 \geq \lambda_2 \geq \lambda_3$ the eigenvalues of $\boldsymbol{D}(\boldsymbol{x})$ for some $\boldsymbol{x} \in \Omega$ and let $\boldsymbol{v}_i$ represent the eigenvector corresponding to λ_i. The diffusion strength therefore assumes its maximum in the direction of $\boldsymbol{v}_1$ and is proportional to λ_1. Hence, $\boldsymbol{v}_1$ may also represent the tangential direction of the possible neural tract at the point $\boldsymbol{x}$. The number of neural fibers actually present at this location may be considered proportional to the anisotropy strength, which is quantified by the *fractional anisotropy* (FA, see e.g. [4]) defined as

$$FA = \frac{\sqrt{3\left((\lambda_1-\lambda)^2+(\lambda_2-\lambda)^2+(\lambda_3-\lambda)^2\right)}}{\sqrt{2\left(\lambda_1^2+\lambda_2^2+\lambda_3^2\right)}}, \tag{2.1}$$

where

$$\lambda = \frac{1}{3}\left(\lambda_1+\lambda_2+\lambda_3\right).$$

It is easy to verify that $FA \in [0,1)$, where 0 indicates perfect isotropy (one should not expect any anisotropic structures, i.e. fibers, at the point $\boldsymbol{x}$) and 1 would mean perfect anisotropy ($\lambda_2 = \lambda_3 = 0$, Γ degenerates to a line segment). Generally, the greater the value of FA, the more neural fibers are present. However, the converse does not hold: As the diffusion ellipsoid is a quadric surface, it cannot represent the focusing of anisotropy to more than one main direction (e.g. fiber bundle crossing). In such a case, FA would approach zero.

Even though the idea was to use the original tensor field $\boldsymbol{D}$ for the visualization process, the anisotropy strength described by $\boldsymbol{D}$ has proved to be too weak to produce observable streamlines. To overcome this difficulty, a preprocessing utility has been created to modify $\boldsymbol{D}$ so that the corresponding diffusion ellipsoids are stretched along their largest principal axis. For each voxel $\boldsymbol{x}$, the positive eigenvalues λ_i are calculated by an explicit formula for finding the roots of the characteristic polynomial of $\boldsymbol{D}(\boldsymbol{x})$, operating in $\mathbb{R}$. Afterwards, the symmetry of $\boldsymbol{D}(\boldsymbol{x})$ is used with advantage to find an orthonormal set of eigenvectors $\boldsymbol{v}_i$. The fractional anisotropy and the modified tensor field $\tilde{\boldsymbol{D}}$ are then computed and saved together to a single NetCDF dataset.

Anisotropic diffusion by the Allen-Cahn equation. When the tensor field is ready, the actual visualization phase takes place. Generally, the diffusion process found in various contexts can be described by a mathematical model formulated as a problem for a partial differential equation with a diffusion term [3]. For the purposes of the proposed algorithm, the Allen-Cahn equation [2] has been chosen.

Consider the time interval $\mathcal{J} = (0,T)$, the domain $\Omega \subset \mathbb{R}^3$ in the form of a block and the diffusion tensor field $\tilde{\boldsymbol{D}} : \bar{\Omega} \mapsto \mathbb{R}^{3\times 3}$ representing the input data. The initial boundary value problem for the Allen-Cahn diffusion equation reads

$$\xi\frac{\partial p}{\partial t} = \xi\nabla\cdot\tilde{\boldsymbol{D}}\nabla p + \frac{1}{\xi}f_0(p) \qquad \text{in } \mathcal{J}\times\Omega, \tag{2.2}$$

$$\left.\frac{\partial p}{\partial n}\right|_{\partial\Omega} = 0 \qquad \text{on } \mathcal{J}\times\partial\Omega, \tag{2.3}$$

$$\left.p\right|_{t=0} = I \qquad \text{in } \Omega, \tag{2.4}$$

where p is the unknown function $p : \bar{\mathcal{J}}\times\bar{\Omega} \to \mathbb{R}$ interpreted as the texture intensity, I represents a noisy initial condition and $f_0(p) = p(1-p)\left(p-\frac{1}{2}\right)$ acts efficiently as a contrast increasing term provided that the small parameter $\xi > 0$ and the final time T are chosen appropriately (in our case by experiment). For the original physical interpretation of f_0 and ξ, see e.g. [1].

The problem (2.2-2.4) is solved numerically on a structured rectangular grid, which will be discussed in more detail in the next section. In principle, the procedure consists of the following steps:

1. The tensor field is interpolated from the original voxel grid onto the computation grid, which is finer in order to achieve greater resolution of the resulting streamlines. Trilinear elementwise interpolation is employed.

2. The initial condition containing random impulse noise is generated on the computational grid.

3. The numerical solution of the given problem is found. The solution is a function of both space and time and its value at some final time $T > 0$ is considered the visualization result.

Colorization and slicing. After the diffusion process is completed, the postprocessing phase begins, involving slicing and colorization. The 3D grid is divided into slices cut in one of the principal planes of the human body: *transverse*, *sagittal*, or *coronal* plane [7]. These planar grayscale images are then colorized by multiplying the brightness of each pixel by the color representation of FA at the corresponding voxel of the domain Ω_0 (see Figure 5.3). The color is obtained by using a linear function mapping the interval

$$\left[0, \max_{\boldsymbol{x}\in\Omega_0} FA\left(\boldsymbol{x}\right)\right]$$

onto the color scale. The NetCDF dataset produced by performing the diffusion process can be reused to generate several sets of slices.

3 Numerical solution of the problem for the Allen-Cahn equation

For numerical solution, the *method of lines* [10] is utilized. Applying a finite volume discretization scheme in space, the problem (2.2-2.4) is converted to a system of ODE in the form of a semidiscrete scheme

$$\xi \frac{\mathrm{d}}{\mathrm{d}t} p_K\left(t\right) = \xi \sum_{\sigma\in\mathcal{E}_K} F_{K,\sigma}\left(t\right) + \frac{1}{\xi} f_{0,K}\left(t\right) \qquad \forall K \in \mathcal{T} \tag{3.1}$$

where $\mathcal{T}$ is an admissible finite volume mesh [5], $K \in \mathcal{T}$ is one particular control volume (cell) and $\mathcal{E}_K$ is the set of all faces of the cell K. $F_{K,\sigma}\left(t\right)$ represent the respective numerical fluxes at the time t, which contain difference quotients approximating the derivatives $\partial_x p$, $\partial_y p$, $\partial_z p$ at the center of the face σ. To solve (3.1), we employ the 4th order Runge-Kutta-Merson solver with adaptive time stepping.

Artificial dissipation and finite volume scheme design. One can assess the behavior of the numerical solution with respect to *artificial (numerical) dissipation* depending on the exact form of $F_{K,\sigma}$. This phenomenon demonstrating itself as an additional

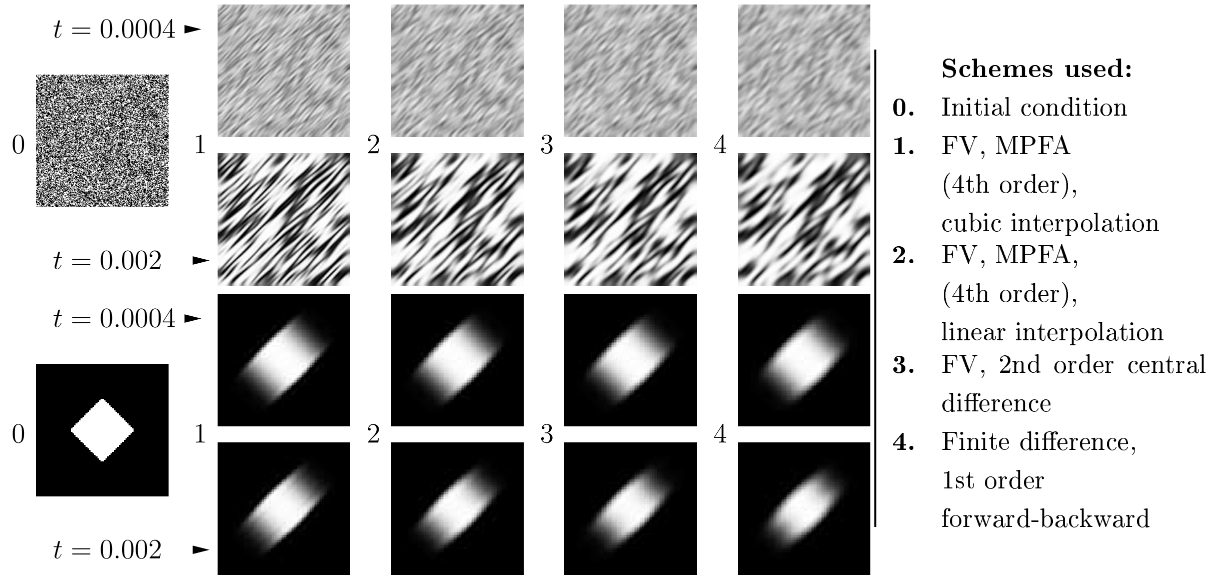


Figure 3.1: Artificial diffusion in different numerical schemes. 2 time levels for 2 different initial conditions.

isotropic diffusion may significantly deteriorate the visual quality of the result. This is because the streamlines emerging in the solution are thin high frequency structures. To be treated correctly, they require the difference operators used in $F_{K,\sigma}$ to be of an appropriate order [13, 8].

Having the results obtained using different schemes available, one can decide on the best of them by mere visual comparison. We have compared finite volume schemes based on three different discretizations of $F_{K,\sigma}$ together with a standard 1st order forward-backward finite difference scheme. The comparison performed in two different settings was restricted to $\mathbb{R}^2$ and is shown in Figure 3.1. In both cases, the initial condition depicted on the very left underwent a process of anisotropic diffusion directed along the axis $y = x$. Least artificial dissipation was produced by the *multipoint flux approximation* (MPFA) scheme where the numerical flux $F_{K,\sigma}$ was obtained using the rules below:

- The difference quotient approximating the derivative in the direction perpendicular to the face σ uses a non-equidistant point distribution in order to avoid redundant interpolation (Figure 3.2a). Its 1-dimensional analog for a function $u \in \mathrm{C}^1(\mathbb{R})$ can be represented by the formula

$$\left.\frac{\mathrm{d}u}{\mathrm{d}x}\right|_{x_{i+\frac{1}{2}}} \approx \frac{1}{24h}\left(u_{i-1} - 27u_i + 27u_{i+1} - u_{i+2}\right)$$

where $x_j = j \cdot h$, $u_j = u(x_j)$ for $j \in \mathbb{Z}, h > 0$.

- The remaining derivatives are approximated using a uniform 5-point stencil. Again, its 1D analog can be written as

$$\left.\frac{\mathrm{d}u}{\mathrm{d}x}\right|_{x_i} \approx \frac{1}{12h}\left(u_{i-2} - 8u_{i-1} + 8u_{i+1} - u_{i+2}\right).$$

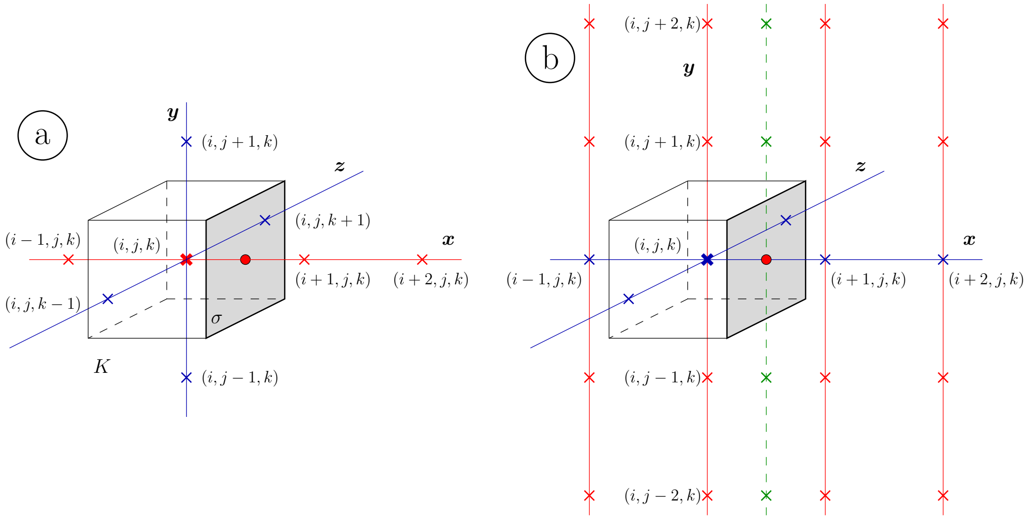


Figure 3.2: Point stencils of difference quotients for derivative approximations in the MPFA finite volume scheme.

Moreover, the stencil points (the crosses along the dashed line in Figure 3.2b) are interpolated from the neighboring grid nodes using 1-dimensional cubic interpolation.

4 Convergence properties

We have been dealing with the derivation of the error estimate for a general finite volume scheme with first order flux approximation on a general mesh. For all $K \in \mathcal{T}$, denote by p_K^n the value obtained by numerical solution of 3.1 approximating $p\left(x_K, nk\right)$ where $x_K \in K$. The pointwise error is then given by

$$e_K^n = p\left(x_K, t_n\right) - p_K^n.$$

The goal is to prove a first order error bound

$$\sqrt{\sum_{K \in \mathcal{T}} \left(e_K^n\right)^2 m(K)} \leq C\left(h + k\right) \qquad \forall n, nk < T$$

so far available for the isotropic case and a special centered difference scheme only (see also [5]). Here we only discuss the experimentally measured convergence rates, suggesting the limitations and possibilities of the theoretical error estimate.

The *experimental order of convergence* (EOC) is obtained by computing the solution on a sequence of gradually refining grids and is defined as

$$\mathrm{EOC}_i = \log\left(\frac{\mathrm{Error}_i}{\mathrm{Error}_{i-1}}\right) \Bigg/ \log\left(\frac{h_i}{h_{i-1}}\right),$$

h	$\mathrm{L}_\infty(\mathcal{J};\mathrm{L}_2(\Omega))$ error $\times 10^{-4}$	EOC in $\mathrm{L}_\infty(\mathcal{J};\mathrm{L}_2(\Omega))$	$\mathrm{L}_\infty(\mathcal{J};\mathrm{L}_\infty(\Omega))$ error $\times 10^{-3}$	EOC in $\mathrm{L}_\infty(\mathcal{J};\mathrm{L}_\infty(\Omega))$
0.00990	2.5560	-	5.5110	-
0.00497	0.6389	2.015	1.3560	2.038
0.00332	0.2844	2.005	0.6097	1.979
0.00249	0.1601	2.002	0.3431	2.004

Table 4.1: EOC results for the standard central difference scheme.

h	$\mathrm{L}_\infty(\mathcal{J};\mathrm{L}_2(\Omega))$ error $\times 10^{-3}$	EOC in $\mathrm{L}_\infty(\mathcal{J};\mathrm{L}_2(\Omega))$	$\mathrm{L}_\infty(\mathcal{J};\mathrm{L}_\infty(\Omega))$ error $\times 10^{-2}$	EOC in $\mathrm{L}_\infty(\mathcal{J};\mathrm{L}_\infty(\Omega))$
0.00971	3.2350	-	2.2190	-
0.00493	1.6190	1.021	1.1140	1.016
0.00330	1.0790	1.012	0.7440	1.008
0.00248	0.8095	1.008	0.5585	1.005

Table 4.2: EOC results for the MPFA scheme.

where $h = \max_K \operatorname{diam}(K)$ is the mesh size and Error_i is the difference of the i-th solution from the precise (analytical) solution measured in an appropriate norm. To be able to calculate the analytical solution, we modify the right hand side of (2.2) to obtain an alternate problem with any prescribed solution of class $\mathrm{C}^2\left(\bar{\Omega} \times \mathcal{J}\right)$. Of course, the prescribed solution must satisfy the initial and boundary condition. For two of the schemes compared in Figure 3.1, the results of the experimental convergence analysis for $\boldsymbol{D}$ constant are summarized in Tables 4.1 and 4.2.

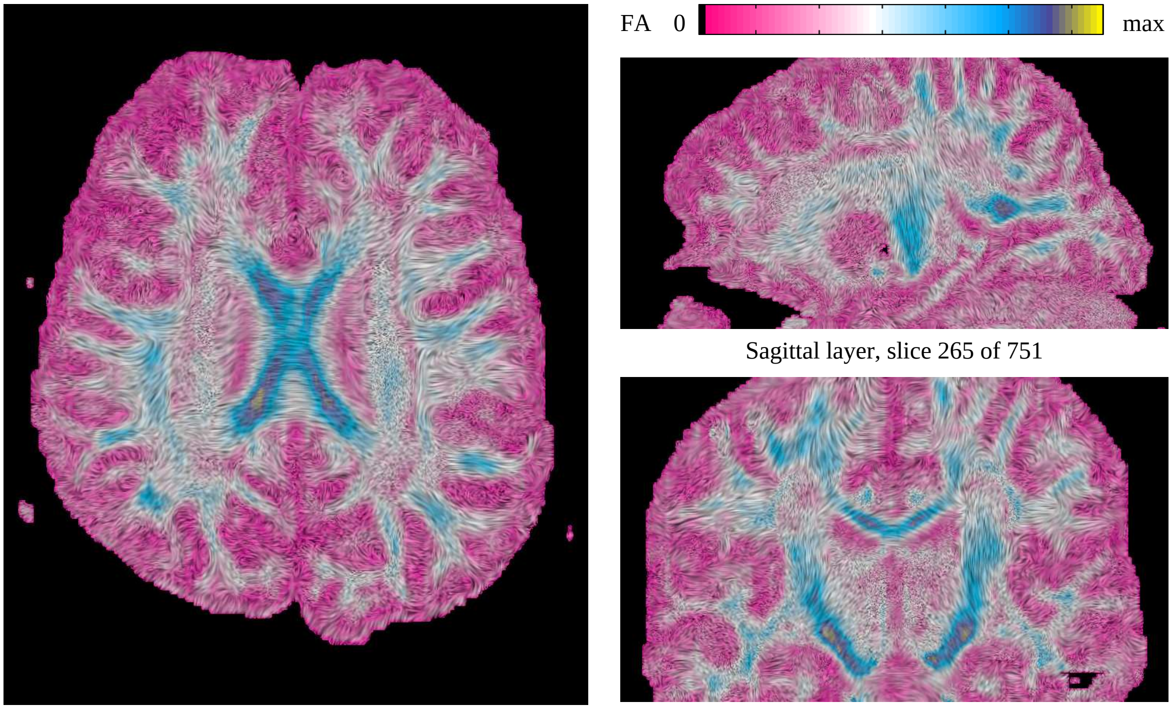
5 Visualization results

In Figures 5.1 and 5.2, we demonstrate the function of the MEGIDDO visualization kit on two sample input datasets. The streamlines of the tensor field indicate the location and direction of the neural tracts. Colorization by the value of fractional anisotropy FA is obtained by performing the color mapping procedure depicted schematically in Figure 5.3 and explained in detail in the last paragraph of Section 2.

Tractography in Figure 5.2 depicts pathological morphology in the patient's cerebrum. The corresponding source dataset has been obtained by a modern 3T scanner at IKEM, Prague.

6 Conclusion and additional remarks

We present a fully functional implementation of the DTI visualization procedure based on anisotropic diffusion of a noisy texture. This approach may represent a suitable complement to the established tractography techniques utilizing explicit fiber tracking algorithms. It provides a global overview of the fiber tract structure in the whole brain



Sagittal layer, slice 265 of 751

Transverse layer, slice 200 of 445

Coronal layer, slice 480 of 900

Figure 5.1: Slices of the DTI complete brain visualization, examination of a healthy volunteer. Dimensions: 900 × 751 × 445 voxels.

or in the specified region. The procedure has significant resource demands in terms of memory and CPU time and therefore has been implemented as a parallel algorithm by means of the MPI interface [11].

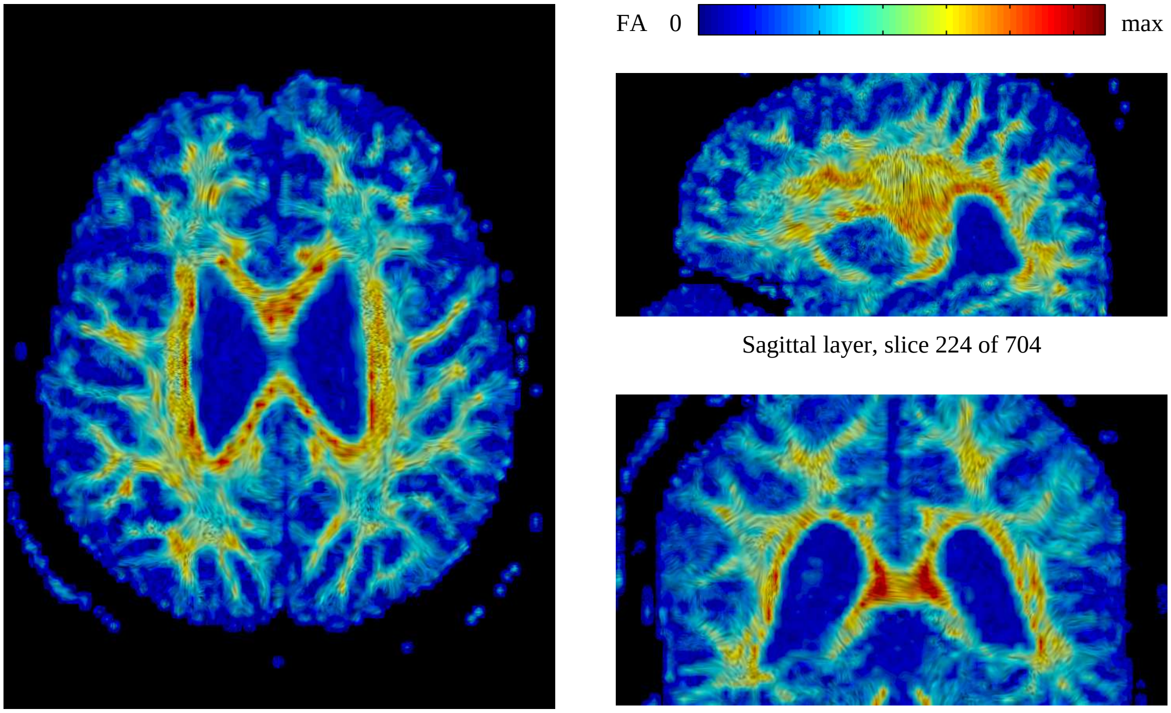
Acknowledgments: This work was carried out under the HPC-EUROPA++ project (project number: 211437), with the support of the European Community - Research Infrastructure Action of the FP7 ”Coordination and support action” Program. Partial support of the project "Jindřich Nečas Center for Mathematical Modeling", No. LC06052. Special thanks to the colleagues at the Institute for Clinical and Experimental Medicine (IKEM) in Prague for providing input datasets, consultations, and support.

References

[1] M. Beneš. *Mathematical analysis of phase-field equations with numerically efficient coupling terms.* Interfaces and Free Boundaries **3** (2001), 201–221.

[2] M. Beneš. *Diffuse-interface treatment of the anisotropic mean-curvature flow.* Applications of Mathematics **48** (2003), 437–453.

[3] M. Beneš, V. Chalupecký, and K. Mikula. *Geometrical image segmentation by the Allen-Cahn equation.* Applied Numerical Mathematics **51** (2004), 187–205.

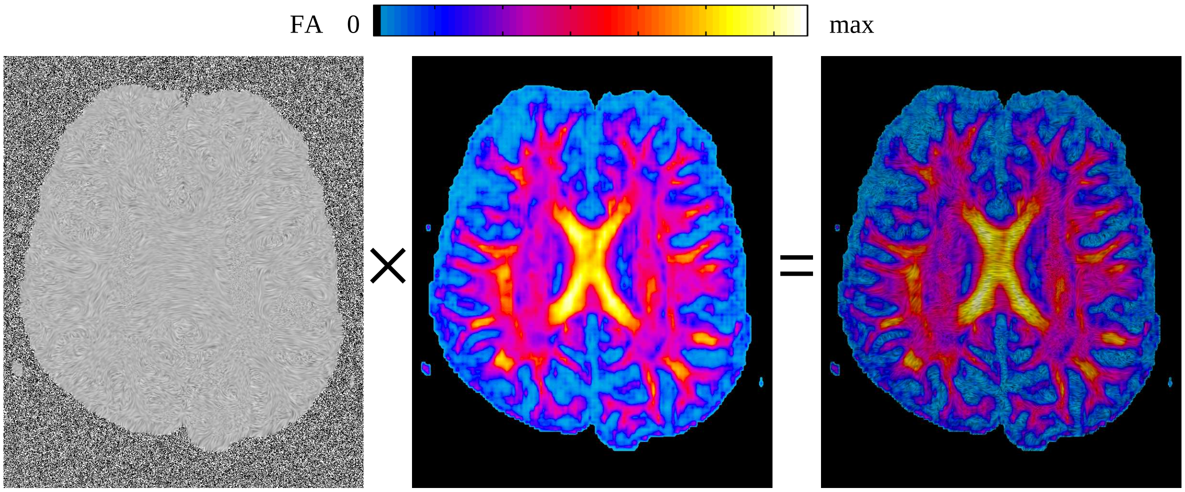


Sagittal layer, slice 224 of 704

Transverse layer, slice 156 of 397

Coronal layer, slice 320 of 900

Figure 5.2: Slices of the DTI complete brain visualization, examination of a patient. Dimensions: 900 × 704 × 397 voxels.



Raw solution slice

FA image using color scale

Final colorized slice

Figure 5.3: Colorization of the raw result of the visualization process.

[4] D. L. Bihan et al. *Diffusion tensor imaging: Concepts and applications.* Journal of Magnetic Resonance Imaging **13** (2001), 534–546.

[5] R. Eymard, T. Gallouët, and R. Herbin. *Finite volume methods.* In 'Handbook of Numerical Analysis', P. G. Ciarlet and J. L. Lions, (eds.), volume 7, Elsevier (2000), 715–1022.

[6] P. Fillard and G. Gerig. Analysis tool for diffusion tensor MRI. In 'Proc. of Medical Image Computing and Computer-Assisted Intervention (MICCAI)', 967–968. Springer, (2003).

[7] W. Kahle, H. Leonhardt, and W. Platzer. *Color Atlas and Textbook of Human Anatomy in 3 Volumes*, volume 1: Locomotor System. Georg Thieme Verlag, Stuttgart, 3rd edition, (1986).

[8] H. Lomax, T. H. Pulliam, and D. W. Zingg. *Fundamentals of Computational Fluid Dynamics.* Springer, (2001).

[9] S. Mori and J. Zhang. *Principles of diffusion tensor imaging and its applications to basic neuroscience research.* Neuron **51** (2006), 527–539.

[10] W. E. Schiesser. *The Numerical Method of Lines: Integration of Partial Differential Equations.* Academic Press, San Diego, (1991).

[11] M. Snir, S. Otto, S. Huss-Ledermann, D. Walker, and J. Dongarra. *The Complete MPI Reference.* The MIT Press, (1995).

[12] P. Strachota. Anisotropic diffusion in mathematical visualization. In 'Science and Supercomputing in Europe - Report 2007', 826–831, Bologna, (2008). CINECA Consorzio Interuniversitario.

[13] P. Strachota. Antidissipative numerical schemes for the anisotropic diffusion operator in problems for the allen-cahn equation. In 'ALGORITMY 2009 - Proceedings of contributed lectures and posters', A. Handlovičová, P. Frolkovič, K. Mikula, and D. Ševčovič, (eds.), volume 18, 134–142. Slovak University of Technology in Bratislava, (2009).

[14] D. Tschumperlé and R. Deriche. Variational frameworks for DT-MRI estimation, regularization and visualization. In 'Ninth IEEE International Conference on Computer Vision (ICCV'03)', volume 1, 116, (2003).

[15] D. Tschumperlé and R. Deriche. Tensor field visualization with PDE's and application to DT-MRI fiber visualization. INRIA Sophia-Antipolis, Odyssée Lab, France, (2004).

[16] C. F. Westin et al. *Processing and visualization for diffusion tensor MRI.* Medical Image Analysis **6** (2002), 93–108.

A New Approach to Estimating the Bellman Function*

Jan Zeman

3rd year of PGS, email: janzeman3@seznam.cz
Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, CTU in Prague
advisor: Tatiana Valentine Guy, Institute of Information Theory and Automation, ASCR

Abstract. The paper concerns an approximate dynamic programming. It deals with a class of tasks, where the optimal strategy on a shorter horizon is close to the global optimal strategy. This property leads to a new, specific, design of the Bellman function estimation. The paper introduces the proposed approach and provides an illustrative example performed on the futures trading data.

Abstrakt. Článek se zabývá oblastí aproximoveného dynamického programování, konkrétně úlohou, kde optimální strategie pro kratší horizont je blízká nebo stejná jako optimální strategie pro celý horizont. Tato vlastnost vede k nové metodě výpočtu Bellmanovy funkce. Článek pouze uvádí první kroky v práci s daným typem úlohy a vše demonstruje na příkladu obchodování s futures kontrakty.

1 Introduction

The motivation of the research originates from the future trading, with the main aim to design a profitable strategy of buying/selling of commodities, betting on the increase/decrease of the future price [3].

To this aim, the available historical price-data covering 35 markets from the last 15 years has been analyzed. Comparison of the trading strategies designed for different values of time horizon has shown that an increase of amount of data causes only partial change of the strategy designed. Moreover the non-changing part of the strategy is always situated at the beginning and is similar to the best strategy designed for a much larger horizon. This property, specific for futures trading data, has been exploited to design and implement the proposed approach on the approximate dynamic programming.

The paper introduces the mentioned property in more details as well as outlines possible application to dynamic programming.

The dynamic programming is an optimization method based on the idea presented in [1]. The dynamic programming maximizes the gain G over a sequence of decisions $x_t, \ldots, x_T$:

$$\max_{x_t,\ldots,x_T} G, \tag{1}$$

*This work has been supported by cooperation with Colosseum a.s., CNR-IMATI and grants GACR 102/08/0567, MŠMT 2C06001.

where $t \in \{1, \ldots, T\}$ is a discrete time and T is finite, possibly large, horizon. The set $\{1, \ldots, T\}$ is called a decision period.

While dynamic programming searches the argument maximizing $\arg\max_{x_t,\ldots,x_T} G$, the maximum the optimal value can be obtained by maximization $\mathcal{V}_t = \max_{x_t,\ldots,x_T} G$ is characterized by the Bellman function $\mathcal{V}_t$ [1]. The main drawback of dynamic programming is the curse of dimensionality (see [5]), therefore the approximate solutions should be searched for.

This paper contributes at the approximation of Bellman function. The proposed approach is useful for the tasks arising in economic analysis and trading and can be of interest for other applications.

The Section 2.1 introduces the dynamic programming and formulates the Bellman equation. The Section 2.2 deals with the method of a comparison of two strategies, which leads to a design of a system of Bellman equations. The system can be used for an estimation of the Bellman function in a parametric shape (see 2.4). The paper is concluded by an example in Section 3, where the proposed approach is applied to futures trading data.

2 The Field of Interest

2.1 Dynamic programming task

A dynamic programming is an method applicable to the problems when it is necessary to find the best decision one after another. The decision making task assumes a *decision maker* and a *system*. The system is a part of the world, which is of interest for the decision maker. The system can be very complex to be fully characterized, moreover the knowledge about the system is usually partial.

The decision maker has own aim related to the system. The aim are expressed in the form of a *gain function* G_τ^T, which quantifies the degree of reaching the aim on (τ, T). The decision maker applies a sequence of decisions $(x_1, \ldots, x_T)$ to reach his aims, i.e. maximizing his gain function over the decision period:

$$\max_{x_1,\ldots,x_T} G_1^T. \tag{2}$$

The decision maker observes a *system output* $(y_1, \ldots, y_T)$. The information available to the decision maker at time t to design a decision x_t is called *knowledge*. The knowledge $\mathcal{P}_t$ contains a history of the system output and previous decisions: $\mathcal{P}_t = (y_1, \ldots, y_t, x_1, \ldots, x_{t-1})$.

The system and the decision maker form a closed loop. The decision maker enriches his knowledge by system output y_t and designs a decision x_t. The decision can be realized as a system input, which influences the further behavior of the system. This process is repeated at each t up to the horizon T.

At time t, the decision maker maximizes:

$$\max_{x_t,\ldots,x_T} G_t^T. \tag{3}$$

The gain function G_t^T depends on the system output over the whole time horizon $(y_t, \ldots, y_T)$. However the information available to the decision maker at time t is $\mathcal{P}_t$. Therefore the decision maker is forced to use the *expected value*:

$$\mathcal{E}(a|b) = \int_{a \in a^*} af(a|b)da,$$

where $\mathcal{E}(a|b)$ is the expected value of the variable a conditioned on the knowledge of variable b and $f(a|b)$ is the probability density function of a defined at the set a^* and conditioned on b.

Thus, the decision maker maximizes the expected value of the gain at time t:

$$\mathcal{V}(\mathcal{P}_t) = \max_{x_t, \ldots, x_T} \mathcal{E}(G_t^T | \mathcal{P}_t, x_t, \ldots, x_T),$$

which defines the Bellman function $\mathcal{V}(\mathcal{P}_t)$.

The assumption of an additive gain function

$$G_{t_1}^{t_2} = G_{t_1}^{t} + G_{t+1}^{t_2} \quad \text{for } t_1 < t < t_2$$

and the optimality principle [2] allow us to rewrite the Bellman function in the recursive shape:

$$\mathcal{V}(\mathcal{P}_t) = \max_{x_t, \ldots, x_{t+h}} \mathcal{E}(G_t^{t+h} + \mathcal{V}(\mathcal{P}_{t+h+1}) | \mathcal{P}_t, x_t, \ldots, x_{t+h})), \tag{4}$$

where the maximum arguments $x_t, \ldots, x_{t+h}$ are the proposed decisions and h is constant, which allows the design of multi-step decision, its value is connected with shape of gain function or kind of task.

The described formulation is too general for the class of tasks considered in futures trading area, therefore the following assumptions are accepted from here onward:

2.1.1 Discrete decisions

the decisions are chosen from a finite, discrete and predefined set.

2.1.2 Open loop

the decision has no influence on the system.

2.2 Similarity indexes

For each time t, there is a system output sequence $(y_1, \ldots, y_t)$ available. We design the optimal strategy $X^t = (x_1, \ldots, x_t)$, where we use the time t as horizon. The strategy is optimal only on the time interval $(1, \ldots, t)$, and is denoted by the superscript t.

Designing the strategy X^t at each time t, a sequence of enlarging strategies is obtained:

$$\begin{array}{llll}
\{y_1\} & \Rightarrow & \{x_1^1\} & = & X^1, \\
\{y_1, y_2\} & \Rightarrow & \{x_1^2, x_2^2\} & = & X^2, \\
\{y_1, y_2, y_3\} & \Rightarrow & \{x_1^3, x_2^3, x_3^3\} & = & X^3, \\
 & \vdots & & & \\
\{y_1, \ldots\ldots, y_t\} & \Rightarrow & \{x_1^t, \ldots\ldots, x_t^t\} & = & X^t.
\end{array}$$

Let compare the designed strategies with the longest strategy X^T for $t = T$. The strategy X^T is called the optimal strategy, because it is optimal for the decision period , i.e. $\{1, \ldots, T\}$. The other strategies are called suboptimal strategies, because they are not optimal for the whole decision period, but only for the respective sub-periods.

Let us assume that the suboptimal strategies X^t converges to the optimal strategy X^T with the growing t and let take the first t elements of the strategy X^T.

Now we can compare two sequences: $(x_1^T, \ldots, x_t^T)$, which is the beginning part of the optimal strategy X^T and $(x_1^t, \ldots, x_t^t)$, which is the suboptimal strategy designed at time t. To compare these sequences, we used the following similarity indexes:

- *Similarity index S_t :*

  $$S_t = \sum_{i=1}^{t} \delta(x_i^t, x_i^T), \tag{5}$$

  where $\delta(x, y) = 1$ for $x = y$ and $\delta(x, y) = 0$ for $x \neq y$.
  The similarity index S_t is a number of identical elements in the sequences $(x_1^T, \ldots, x_t^T)$ and $(x_1^t, \ldots, x_t^t)$.

- *Strict similarity index S_t :*

  $$s_t = \max_i \{i; (\forall j \in \mathcal{N})(j \leq i \Rightarrow x_j^t = x_j^T)\}. \tag{6}$$

  The strict similarity index is the maximal length of the non-broken identical subsequence beginning by the first element.

The definitions of S_t and s_t imply $s_t \leq S_t \leq t$.

To illustrate the introduces notions, let us consider the following suboptimal and optimal strategies: $$\begin{array}{rl} X^t = & \{\,1 \;\; 1 \;\; 1 \;\; 1 \;\; 0 \;\; 1 \;\; 1 \;\; 0 \ldots 0\,\}, \\ X^T = & \{\,1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \ldots 1\,\}, \end{array}$$
where the sequence X^T is cut to have the same length as X^t. Sequences have 4 elements identical, the fifth element differs, the sixth and seventh elements are identical and then sequences differ.

There, the similarity index $S_t = 6$, because there are 6 identical elements in the sequences. The strict similarity index $s_t = 4$, because the fourth element is the last element, before the first difference occurs.

2.3 Bellman equation and similarity indexes

The solution of Bellman equation (4) is the most important part of the dynamic programming task. The term 'solution' means finding the Bellman function, a task that can be very complex due to the backward recursive shape of the equation. The optimal actions are only by-products of this solution.

The use of similarity indexes s_t and S_t could is useful, if they grow with the time $s_t \approx t$, $S_t \approx t$.

At each time t, the sequence of optimal actions of length s_t is known and the set of the Bellman equations is:

$$\mathcal{V}(\mathcal{P}_k) = \max_{x_k, \ldots, x_{k+h}} \mathcal{E}(G_k^{k+h} + \mathcal{V}(\mathcal{P}_{k+h+1})|\mathcal{P}_k, x_k, \ldots x_{k+h})), \tag{7}$$

where $k \in \{1, \dots, s_t - h\}$.

The maximization can be carried out by substitution of suboptimal actions $X^t = (x_1, \dots, x_{s_t})$:

$$\begin{aligned} \mathcal{V}(\mathcal{P}_k) &= \mathcal{E}(G_k^{k+h} + \mathcal{V}(\mathcal{P}_{k+h+1})|\mathcal{P}_k, x_k^t, \dots, x_{k+h}^t)) \\ &\quad \text{for} \quad k \in \{1, \dots, s_t - h\}. \end{aligned} \tag{8}$$

Due to $k \leq t$, the expected values converges to substitution of known values $\mathcal{P}_k$, $(x_k^t, \dots, x_{k+h}^t)$. Thus, the system of functional equations (8) should be solved to obtain the Bellman function.

2.4 Parametric shape of Bellman function

A lot of technical details should be resolved before full use of the described approach. We restrict the design to parameterized form of Bellman function:

$$\mathcal{V}(\mathcal{P}_t) \approx V(\mathcal{P}_t; \Theta), \tag{9}$$

where $\Theta \in \Theta^*$ is a vector of unknown parameters. Then, the solution of the Bellman equation converges to estimation of the parameters Θ and data prediction. Inserting (9) into the system of equations (8), one can write:

$$\begin{aligned} V(\mathcal{P}_k; \Theta) + \kappa_k &= \mathcal{E}(G_k^{k+h} + V\big(\mathcal{P}_{k+h+1}; \Theta)|\mathcal{P}_k, x_k^t, \dots, x_{k+h}^t\big), \\ &\quad \text{for} \quad k \in \{1, \dots, s_t - h\}. \end{aligned} \tag{10}$$

where κ_k is an error caused by approximation.

The system of functional equations (8) is further reduced to the system of algebraic equation (10).

2.5 Task classification

Presented design assumes that s_t grows approximately with time t. This is, of course, only the ideal case. Generally there are three types of tasks:

- Task with a strong similarity - is a task, where s_t and S_t grow with the time. Therefore, the number of equations in system (8) or (10) grows with t. Thus, the presented design can be applied.

  In case of use parameterized shape and system (10), it can happen that the number of independent equations overgrows the degree of freedom and the desired solution should be searched respecting that.

- General task without a similarity - where s_t and S_t are small constants independent of t. In this case, the system has a small number of equations. The number of equations in (8) and (10) do not grow, or grow by jumps. There could not be enough equations to find a solution. In this case, different design of the Bellman function should be used. However even the available "poor" system of equations can be used as a prior information about the Bellman function.

- Task with a weak similarity - where s_t is a small constant or growing only by jumps, but S_t grows with t. The proposed approach can be used, but systems (8) and (10) must be written for $k \in \{1, \ldots, S_t - h\}$.

  The approach can be applied carefully not all - but almost all - equations in systems (8) and (10) are valid. Thus the design systematically uses invalid equations and this must be respected.

2.6 Causality problem

Presented classification is non-causal, because the optimal strategy X^T, designed over all decision period should be known for the calculation of s_t and S_t and the approach can be used for off-line experiments only.

On-line use needs to study the behavior of sequences of the suboptimal strategies $X^1, X^2, \ldots, X^t$ and to estimate the value of s_t.

3 Example: Futures Trading

Futures trading task is a task typically solved by exchange speculators, who know the past price sequence and try to decide, whether to buy or sell an object of interest. A profit is made, when the speculator guesses the direction of the price evolution, otherwise the speculator loses.

3.1 Futures trading as a game

From out point of view, the futures trading task can be interpreted as turn based game: The player obtains a price y_t at the beginning of each turn $t \in \{1, 2, \ldots, T\}$. He chooses his decision x_t, whether the price should increase $x_t = 1$ or decrease $x_t = -1$, or player can decide not to play for the turn $x_t = 0$. If player changes the choose x_t according to previous decision x_{t-1}, then he pays a transaction cost $C|x_{t-1} - x_t|$. At the beginning of next turn $t + 1$, the player makes profit of $(y_{t+1} - y_t)x_t$, therefore when player bets the right way, he makes money, otherwise he loses.

The player tries to maximize his profit up to horizon T:

$$G_1^T = \sum_{t=1}^{T} (y_t - y_{t-1})x_{t-1} - C|x_{t-1} - x_t|.$$

The initial decision is necessary to be defined as $x_0 = 0$.

The described game is a typical optimization problem of dynamic programming (see [2]) and as such it should be solved.

3.2 Similarity indexes

It is useful to characterize the systems (8) and (10) according the time t, instead of $k \in \{1, \ldots, s_t - h\}$. Thus, we calculate following constants:

$$c_1 = \max_{t \in \{1 \ldots T\}} (t - s_t), \tag{11}$$

$$c_2 = \max_{t \in \{1 \ldots T\}} (t - S_t), \tag{12}$$

and characterize the systems (8) and (10), which is subset of the original set of equations.

The constants c_1, c_2 characterize maximal number of non-optimal decisions in X^t, which is related with the risk of usage the invalid equations in systems of equations (8) and (10). Hence, the less value of c_1, c_2 is better.

The causal estimation of similarity indexes can be done by analyzing differences between the two suboptimal strategies X^{t-1} and X^t , cf. (5) and (6):

$$\hat{S}_t = \sum_{i=1}^{t-1} \delta(x_i^{t-1}, x_i^t), \tag{13}$$

$$\hat{s}_t = \max_i \{i; (\forall j \in \mathcal{N})(j \leq i \Rightarrow x_j^{t-1} = x_j^t)\}. \tag{14}$$

Analogically can be obtained causal estimation of the constants c_1 and c_2 at the time t:

$$\hat{c}_{1,t} = \max_{i \in \{1,\ldots,t\}} (i - \hat{s}_i), \tag{15}$$

$$\hat{c}_{2,t} = \max_{i \in \{1,\ldots,t\}} (i - \hat{S}_i), \tag{16}$$

The final value of $\hat{c}_{1,t}$ and $\hat{c}_{2,t}$ is not so important as their behavior at time $t < T$. The values of $\hat{c}_{1,t}$ and $\hat{c}_{2,t}$ increase with the time t. It is expected that their values converge to a small constant, which is reached very early, therefore the time of the last change $t_{ch;1}$ and $t_{ch;2}$ is documented.

We have 35 price sequences available for the offline experiments. The data were collected once a day, when the exchange was closing, each data set contains data from 1990 to 2005, which makes about 4000 samples all together. Five price sequences were chosen as a representative for the further experiments: Cocoa - CSCE (CC), Petroleum-Crude Oil Light - NMX (CL), 5-Year U.S. Treasury Note - CBT (FV2), Japanese Yen - CME (JY) and Wheat - CBT (W). All constants defined above were estimated for the five reference markets (see Tab. 1).

The table shows good results, because the constants c_1 and c_2 are the same and $c_1, c_2 \ll T$. Moreover, four of sequences have c_1 equal to c_2 for each t, which implies that s_t is equal to S_t. The values of $t_{ch;1}$ and $t_{ch;2}$ show the expected fact, that the values of $\hat{c}_{1,t}$ and $\hat{c}_{2,t}$ do not change often and the causal estimation of $\hat{c}_{1,t}$ and $\hat{c}_{2,t}$ gives satisfactory results near to non-causal values. All these facts led to a conclusion that futures trading task is the task with a strong similarity, as was described in Sec. 2.5.

The exception with a weak similarity is the market with ticker CL. The obtained similarity indexes are depicted in Fig. 1 and Fig. 2. The difference between s_t and t is markable but it has only a local character, therefore the approach can be used - with the expectation of worse results related to the intervals with a weak similarity.

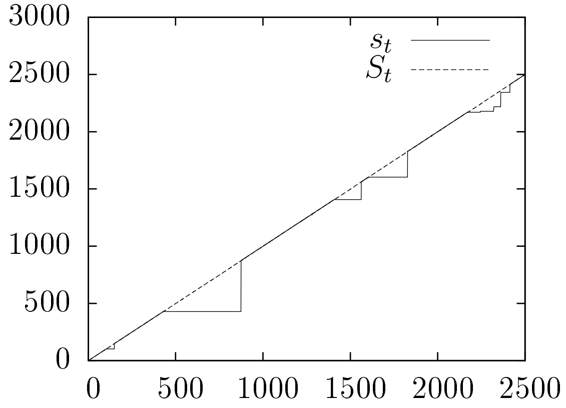


Figure 1: Example of similarity indexes S_t and s_t for CL

3.3 Estimation of Bellman function parameters

Let the parametrized form of Bellman's function be:

$$\mathcal{V}(\mathcal{P}_t) \approx g(x_t)\Psi_t, \tag{17}$$

where $\Psi_t = (y_t, y_{t-1}, \dots, y_{t-n})^T$ is regressor and $g(x_t)$ is a row vector function.

For illustration purpose, the admissible values for x_t are chosen from a set $x^* = \{-1, 0, 1\}$. Thus, the vector function $g(x_t)$ is fully characterized by $3(n+1)$ parameters, which are the elements of vector Θ introduced in Section 2.4. We denote $g(x_t) = (\Theta_{x_t,1}, \Theta_{x_t,2}, \dots, \Theta_{x_t,n+1})$. Each element $\Theta_{x_t,i}$ is a function of x_t. Due to the chosen set x^*, the function $\Theta_{x_t,i}$ is fully characterized by three values.

Substituting (17) into (10), we obtain:

$$g(x_k^t)\Psi_k - g(x_{k+h+1}^t)\Psi_{k+h+1} = G_k^{k+h} - \kappa_k, \tag{18}$$

for

$$k \in \{1, \dots, t - c_1 - h\},$$

Table 1: Dominating constants c_1 and c_2

Market	c_1	c_2	$\hat{c}_{1,T}$	$\hat{c}_{2,T}$	$t_{ch;1}$	$t_{ch;2}$	T
CC	6	6	7	6	342	342	3822
CL	444	6	446	5	847	2205	3863
FV2	8	8	9	8	383	383	3766
JY	4	4	5	4	50	50	3871
W	7	7	8	7	2452	2452	3822

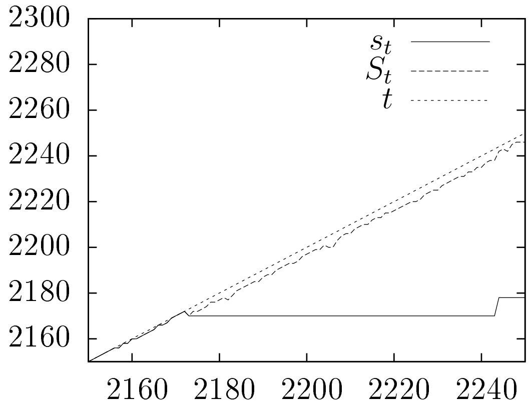


Figure 2: Example of similarity indexes S_t and s_t for CL (detail)

we get a system of linear equations

$$Ax = b - \mathcal{K} \tag{19}$$

where

$$\begin{aligned} x \;=\; & (\Theta_{-1,1}, \ldots, \Theta_{-1,n+1}, \Theta_{0,1}, \ldots, \Theta_{0,n+1}, \\ & \Theta_{1,1}, \ldots, \Theta_{1,n+1}). \end{aligned}$$

and $\mathcal{K} = (\kappa_1, \kappa_2, \ldots, \kappa_{t-c_1-h})$.

The system of linear equations must be solved for each time t to obtain the estimation of the Bellman function values. The number of equations in the system increases by one in each time step. Due to the approximation of the Bellman function, the system need not to be solvable, when the number of equations grows over some threshold. And, an approximate solution of system should be searched. We have applied least square method to minimize the vector of approximation errors $\mathcal{K}$.

Table 2: Results of experiment

Market	MPC	IST
CC	-6 450	-1 490
CL	-12 350	3 390
FV2	-5 701	10 727
JY	-26 568	-35 247
W	-9 792	-1 923

3.4 The results

The obtained parameters are inserted into the parametrized form (17), which is used for maximization of (4). This method corresponds with iterations spread in time (IST) see [4]. To calculate the expected gain, causal predictions generated by autoregressive model were used (see [4]).

As a reference, the results calculated via model predictive control (MPC) were used. The predictive model and task setup were the same for IST.

Final results are summarized in Tab. 2. Presented IST method reaches better results than MPC method at four of the five datasets. Neither MPC nor IST gave enough good results satisfactory to the use for real trading. However, the results obtained by IST are slightly better.

4 Conclusion

The proposed design of the Bellman function is based on searching and analyzing of suboptimal strategies based on known data. The design leads to system of functional equations, but using parametrized shape of Bellman function, the system can be transformed to a system of algebraic equations.

The main idea is to analyze, if the suboptimal strategy contains at least part of the optimal strategy. The task with this property can be either strong or weak similarity. The paper deals with a problem of causal and non-causal analysis leading to a decision which kind of similarity the task exhibits.

The approach is applied and demonstrated on an example of futures trading, which is a typical economic decision making task. The kind of similarity is tested and the behavior of tested method is presented. Then, the new design of Bellman function is applied. Results of experiments are presented and compared to the results of a MPC method and are slightly better.

References

[1] R. Bellman. *Dynamic Programming.* Princeton University Press, Princeton, New Jersey, (1957).

[2] D. Bertsekas. *Dynamic Programming and Optimal Control.* Athena Scientific, Nashua, US, (2001). 2nd edition.

[3] J. Hull. *Options, futures, and other derivatives.* Pearson/Prentice Hall, (2006).

[4] M. Kárný, B. J., T. V. Guy, L. Jirsa, I. Nagy, P. Nedoma, and L. Tesař. *Optimized Bayesian Dynamic Advising: Theory and Algorithms.* Springer, London, (2005).

[5] W. B. Powell. *Approximate Dynamic Programming.* Wiley-Interscience, (2007).

Efficient Scheduling of Data Transfers in Distributed Environment*

Michal Zerola

2nd year of PGS, email: michal.zerola@ujf.cas.cz
Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, CTU in Prague
advisor: Michal Šumbera, Nuclear Physics Institute, ASCR
Jérôme Lauret, Brookhaven National Laboratory, USA
Roman Barták, Faculty of Mathematics and Physics, Charles University

Abstract. Efficient data transfers and placements are paramount to optimizing geographically distributed resources and minimizing the time data intensive experiments's processing tasks would take. We present a technique for planning data transfers to multiple destinations in multi-user environment. We explain the architecture, design and concept of the components of the automated system. The constrained based planning technique, which we have studied and explained in our previous work is outlined in the last section. After the early tests and evaluations in the real world the concept and chosen approach seems to be promising.

Abstrakt. Efektívne dátové prenosy a rozmiestnenia sú kľúčovým prvkom pri optimalizovaní geograficky rozmiestnených zdrojov a zároveň minimalizovaní času, ktorý vyžadujú dátovo intenzívne úlohy experimentov. Predstavíme techniku na plánovanie dátových prenosov do skupiny destinácií vo viac užívateľskom prostredí. Vysvetlíme princíp architektúry, design a koncept jednotlivých komponent automatizovaného systému. Plánovacia technika založená na podmienkach, ktorú sme študovali a prezentovali v našich minulých prácach je načrtnutá v poslednej časti textu. Po prvých testoch a vyhodnoteniach implementácie v reálnom prostredí sa koncept a zvolený princíp zdajú byť sľubné.

1 Introduction

1.1 Problem area

Computationally challenging experiments such as the one from the High Energy and Nuclear Physics (HENP) community have developed a distributed computing approach to face the massive needs of their Peta-scale experiments. The era of data intensive computing has surely opened a vast arena for computer scientists to resolve practical and exciting problems. One of such HENP experiments is the STAR [5] experiment located at the Brookhaven National Laboratory, USA.

In addition to a typical Peta-scale data challenge and large computational needs, this experiment, as a running experiment acquires a new set of valuable real data every year, introducing other dimension of safe data transfer to the problem. From the yearly data

*The investigations have been partially supported by the IRP AVOZ 10480505, by the Grant Agency of the Czech Republic under Contract No. 202/07/0079 and 205-13/201457, by the grant LC07048 of the Ministry of Education of the Czech Republic and by the U.S. Department Of Energy.

sets, the experiment may produce many physics-ready derived data sets which differ in accuracy as the problem is better understood as time passes. Thus, demands for a large-scaled storage management [6] and efficient scheme to distribute data grows as a function of time, while on the other hand, end-users may need to access data sets from previous years at any point in time. Coordination is needed to avoid random access destroying efficiency due to the sharing of common infrastructure.

This includes replication/distribution of centrally acquired data to other computing sites with an emphasis on efficient further processing. Even at the level of a given site, several storage services exist and it is not always all clear on where the files/datasets should be taken from. In this paper we focus on one block of this complex task which is of immediate need by the physicists: "how to bring the desired datasets to the requested destinations in a shortest time?" This problem can be addressed as multiple path planning with shared links and minimizing makespan. Assuming the files from the requested dataset are replicated at several sites and their services, the aim is to select transfer paths for atomic chunks (files) commonly sharing the links/services together with a limited bandwidth and an objective to minimize the makespan. In other words, we want a unique capability of the system to tell from **which services** to grab what portion of requested files and **which transfer path** to use in any time.

In this paper we will concentrate on overview of the design, architecture and important aspects of implementation of the framework. Last but not least, we will describe the planner, brain of the system and underlying mathematical model.

1.2 Related works

The needs of large-scale data intensive projects arising out of several fields such as bio-informatics (BIRN, BLAST), astronomy (SDSS) or HENP communities (STAR, ALICE) have been the brainteasers for computer scientists for years. Whilst the cost of storage space rapidly decreases and computational power allows scientists to analyze more and more acquired data, appetite for efficiency in Data Grids becomes even more of a prominent need.

Decoupling of job scheduling from data movement was studied by Ranganathan and Foster in [8]. Authors discussed combinations of replication strategies and scheduling algorithms, but not considering the performance of the network. The nature of high-energy physics experiments, where data are centrally acquired, implies that replication to geographically spread sites is a must in order to process data distributively. Intention to access large-scale data remotely over wide-area network has turned out to be highly ineffective and a cause of often sorely traceable troubles.

The authors of [10] proposed and implemented improvements to the Condor, a popular cluster-based distributed computing system. The presented data management architecture is based on exploiting the workflow and utilizing data dependencies between jobs through study of related DAGs. Since the workflow in high-energy data analysis is typically simple and embarrassingly parallel without dependencies between jobs these techniques don't lead to a fundamental optimization in this field.

Sato et al. in [9] and authors of [7] tackled the question of replica placement strategies via mathematical constraints modeling an optimization problem in Grid environment.

Solving approach in [9] is based on integer linear programming while [7] uses Lagrangian relaxation method [1]. The limitation of both models is a characterization of data transfers which neglects possible transfer paths and fetching data from a site in parallel via multiple links possibly leading to the better network utilization.

We focus on this missing component considering wide-area network data transfers pursuing more efficient data movement for multi-site multi-user environment. An initial idea of our presented model originates from Simonis [11] and the proposed constraints for traffic placement problem were expanded primarily on links throughputs and consequently on follow-up transfer allocations in time. One of the immense advantages of the constrained based approach is a gentle augmentation of the model with additional real-life rules.

2 Design

In this section we will outline the most important attributes and features we require from the framework.

2.1 Motivation and requested features

To understand the meaning of *optimal selection* and *utilization* of the *resources* let us suppose the following situation. Two users are requesting the same file in two different destinations (Fig. 1 - left). The planner should consider all possible repositories (in this example HPSS ([12]) or Xrootd service ([4])) of the file, together with available transfer paths from these origins to requested destinations. This consideration includes reasoning about the response and transfer time from services at current site to the local cache (*LAN*) but also about the bandwidth for site-to-site transfers (*WAN*). The returned optimal configuration should provide minimal waiting time for users. In our example the file is planned to be staged from Xrootd service to the local cache at BNL site and after that two copies are being further transferred to the requested destinations. As we can see the configuration with overlapping links was preferred which leads to better utilization of the resources.

The system must be also *adaptive* to the changes and fluctuations of the network, hence we cannot allow creation of plans which execution lasts too long. Instead, the plan should be realized iteratively more often and for smaller *batches* of files. On the other hand it has to consider also the current utilization of the links resulted from previous plans (Fig. 1 - right). In particular, a number of either active or queued files per a link has to be checked and to estimate the time the link will be not available (in the figure represented as Gantt charts) and this information has to be used during reasoning about links.

Since the system is supposed to be deployed in a *multi-user environment*, the *fair-share* operation is required as well. In particular, the framework should allow to adapt any queue-based policy providing the required level of fair-shareness. This is naturally offered by batch attitude, where the selection of files into the current batch can follow the proffered fair-share function.

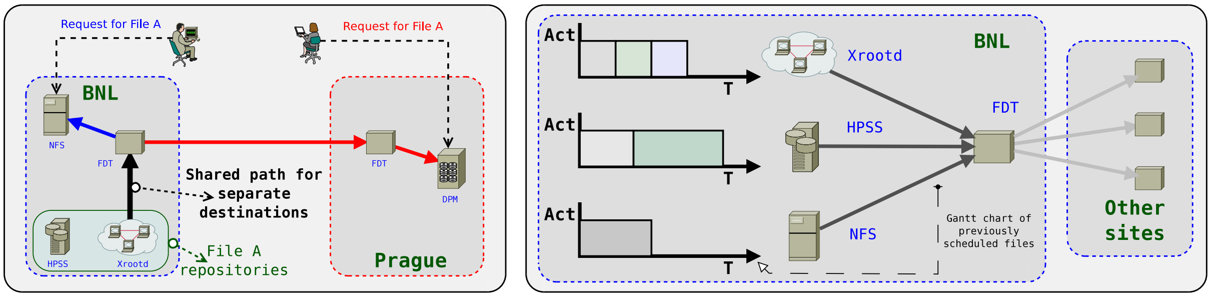


Figure 1: Example of requested features. The left part depicts the plan consisting of paths sharing a link to two different destinations. The right one symbolizes the utilization of the links using Gantt charts.

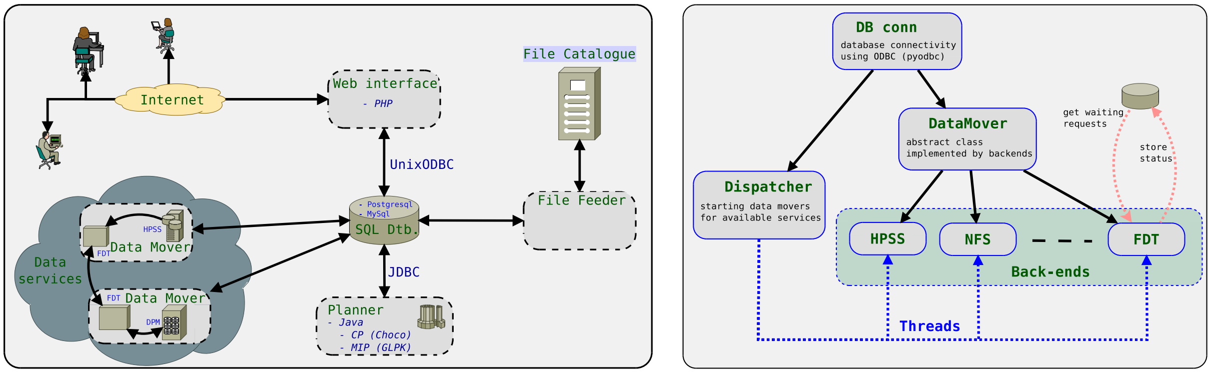


Figure 2: Architecture overview on the left and the scheme of the Data Mover component on the right.

2.2 Architecture

In the Fig. 2 is depicted the scheme of the architecture and concept of the components. To understand their roles we will describe the process flow in the system.

Users enter their requests, which are in the form of *meta-data queries* consisting for instance of a production year or energy selection, via **Web interface**. The queries are stored in a centralized **SQL database** and the database is further populated with a full list of files associated with queries. This is the role of a **Feeder** that contacts the global *File catalog* and stores information about received files, such as available repositories or size. The subset of files is selected according to a queue-based policy and passed to the **Planner** in the next batch. The result from planner, selected transfer path for every file, is stored back to the database.

2.3 Data Mover

Data Mover component is deployed at each computing site and its role is to execute the computed plan for appropriate links. One instance of Data Mover is responsible of transferring files to/from the local data services and of pulling files from remote sites.

The concept of classes (Fig. 2 - right) is following: the *Dispatcher* class starts one thread for each data service (either local or remote). The thread is waiting for the planned transfers with a ‘Queued‘ status on its link by querying the database. As soon as some files are available, the transfer is executed calling appropriate back-end tool. After the either successful or failed transfer the status is updated so other Data Movers are aware of action.

3 Planner

The planner is responsible for generating the transfer paths for a given set of files, which are supposed to be optimal in the sense of minimal makespan as an objective. Our underlying mathematical model is based on constraints and we have studied two solving approaches. First one uses Constraint Programming technique and detailed description can be found in [15], while its search heuristic was published in [14]. The second approach uses Mixed Integer Programming (MIP) method and more detailed description can be found in [13]. Since it provides a fairly better efficiency we will outline this one in the following text.

3.1 Formal model

The problem to be solved and its constraints driven by environmental realities need to formalized using mathematical constraints.

The first part of the input represents the network and file origins. The network, formally a directed weighted graph, consists of a set of nodes $\mathbf{N}$ and a set of directed edges $\mathbf{E}$. The nodes represent the computing sites and the storage elements with extensive access times (e.g. Mass Storage Systems) while the edges transfer links between the nodes. The weight of an edge corresponds to the link bandwidth ($\mathbf{bw}(e)$) between two sites or average latency time for the storage elements (e.g. the time to stage the file from the tape system). The information about file's origins is a mapping of that file to a set of nodes where the file is available.

The second part of the input consists of the requests from the users, namely the set of files that are going to be transferred and their destination sites (a single file can be requested at multiple destinations). The goal of the solver is to produce:

- the transfer paths for each file, i.e. the selection of origins and a valid path starting from the origin node and leading to the destinations such that
- the resulting plan has the minimum makespan (the finish time of the last transfer)

The set $\mathbf{OUT}(n)$ consists of all edges leaving node n, the set $\mathbf{IN}(n)$ of all edges leading to node n. The input received from the users is a set of file names $\mathbf{F}$, where for every file $f \in \mathbf{F}$ we have a set of sources $\mathbf{orig}(f)$ - sites where the file f is already available and a set of destinations $\mathbf{dest}(f)$ - sites where the file f is supposed to be transferred.

The essential idea is to use one decision variable for each file, its destination and edge in a graph. We will refer to this $\{0, 1\}$ variable as X_{fed}, denoting whether file f is routed (value 1) over the edge e of the network or not (value 0) to its destination d. Mathematical

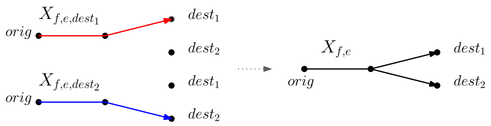


Figure 3: Two independent paths are *glued* together, so the file using their common links will be transferred only once (e.g. the file is staged only once, then transferred to two different destinations)

constraints (1-2), ensuring that if all decision variables have assigned values the resulting configuration contains the independent transfer paths, are analogous to the Kirchhoff's circuit laws.

$$\forall f \in \mathbf{F},\ \forall d \in \mathbf{dest}(f):$$
$$\sum_{e \in \cup \mathbf{OUT}(n | n \in \mathbf{orig}(f))} X_{fed} = 1,\ \sum_{e \in \cup \mathbf{IN}(n | n \in \mathbf{orig}(f))} X_{fed} = 0,\ \sum_{e \in \mathbf{OUT}(d)} X_{fed} = 0,\ \sum_{e \in \mathbf{IN}(d)} X_{fed} = 1 \tag{1}$$

$$\forall f \in \mathbf{F},\ \forall d \in \mathbf{dest}(f),\ \forall n \notin \mathbf{orig}(f) \cup \{d\}:$$
$$\begin{array}{rl} \sum_{e \in \mathbf{OUT}(n)} X_{fed} & \leq 1 \\ \sum_{e \in \mathbf{IN}(n)} X_{fed} & \leq 1 \end{array} \quad \sum_{e \in \mathbf{OUT}(n)} X_{fed} = \sum_{e \in \mathbf{IN}(n)} X_{fed} \tag{2}$$

Having generated all independent paths for a file to each of its destination, we need to *glue* them together. One can look at it as creating a *forest* using the terminology from the graph theory (Figure 3). We achieve it by defining new binary *two-index* variable X_{fe} stating whether file f uses link e (apart from reasoning about destinations).

$$\forall f \in \mathbf{F},\ \forall e \in \mathbf{E},\ \forall d \in \mathbf{dest}(f): X_{fed} \leq X_{fe} \tag{3}$$

$$\forall f \in \mathbf{F},\ \forall e \in \mathbf{E}: \sum_{d \in \mathbf{dest}(d)} X_{fed} \geq X_{fe} \tag{4}$$

$$\forall f \in \mathbf{F},\ \forall n \notin \mathbf{orig}(f) \cup \{d\}: \sum_{e \in \mathbf{IN}(n)} X_{fe} \leq 1 \tag{5}$$

Finally, since we are minimizing the *makespan*, the time to transfer all files to the requested destinations, we define the constraints (6) for estimation of the completition time $\mathbf{T}$ variable and appropriate objective function: *minimize* T.

$$\forall e \in \mathbf{E}: \sum_{f \in \mathbf{F}} \frac{\mathbf{size}(f) \cdot X_{fe}}{\mathbf{bw}(e)} \leq T \tag{6}$$

3.2 Implementation

The model explained in the previous section consists of all linear constraints using *binary* (X) and *real* (T) variables. As explained in [14] for realization of file transfers we do not need an exact schedule, only the plan (the transfer paths) that will be followed by

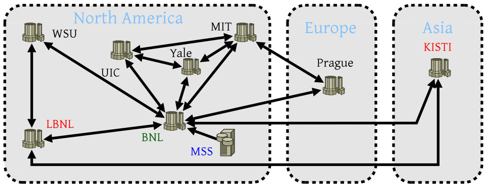


Figure 4: Computing centers in STAR experiment.

Files	**10**	**25**	**50**	**75**	**100**	**200**
Time (s)	0.024	0.258	0.786	1.324	2.518	9.574

Table 1: Average time in seconds to find optimal transfer paths.

the distributed *link managers*. Therefore, after the comparison of solving techniques we chose MIP approach which provides the most efficient results. As the backend MIP solver we use **G**NU **L**inear **P**rogramming **K**it (GLPK [2]) from Java programming language via SWIG interface ([3]).

All presented experiments were performed on laptop with Intel Core2 Duo CPU@1.6GHz, 2GB of RAM, running a Debian GNU Linux operating system. The real-life network structure among *Tier-{0,1,2}* sites in STAR experiment is depicted in Figure 4. The distribution of files is taken from empirical data, where 100% of the files are kept at MSS, 60% at LBNL, 20% at KISTI and 5% are spread among *Tier-2* sites.

According to the results (Table 1) planning in batches of files (to achieve adaptiveness to the network and fair-shareness to the users) seems to be realizable and payed-off by the gained optimality.

4 Conclusions

In this paper we tackle the complex problem of efficient data movements on the network within a distributed environment. The problem itself arises from the real-life needs of the running nuclear physics experiment STAR and its peta-scale requirements for data storage and computational power. We concentrated on the detailed explanation of the requested features from the automated system we have been developing, architecture and concept of the key-stone components. The planning mechanism based on constrained model was presented as discussed in our previously published work. Our main focus is in the implementation of the components, currently being developed and deployed. The early tests in real-world environment point promising indication of correctness of the chosen approach. In the nearest future we will continue with implementation and measurements in the heavy-loaded environment.

References

[1] M. L. Fisher. *The Lagrangian Relaxation Method for Solving Integer Programming Problems*. Management Science **27** (January 1981), 1–18.

[2] GLPK. http://www.gnu.org/software/glpk/.

[3] GLPK-java. http://glpk-java.sourceforge.net/.

[4] A. Hanushevsky, A. Dorigo, and F. Furano. The Next Generation Root File Server. In 'Proceedings of the Computing in High Energy and Nuclear Physics (CHEP) conference', 680–683, (2005).

[5] J. Adams, et al. *Experimental and theoretical challenges in the search for the quark gluon plasma: The STAR collaboration's critical assessment of the evidence from RHIC collisions.* Nuclear Physics A **757** (2005), 102–183.

[6] P. Jakl, J. Lauret, A. Hanushevsky, A. Shoshani, A. Sim, and J. Gu. *Grid data access on widely distributed worker nodes using scalla and SRM.* Journal of Physics Conference Series **119** (July 2008), 072019.

[7] R. M. Rahman, K. Barker, and R. Alhajj. Study of Different Replica Placement and Maintenance Strategies in Data Grid. In 'CCGRID', 171–178. IEEE Computer Society, (2007).

[8] K. Ranganathan and I. Foster. Decoupling Computation and Data Scheduling in Distributed Data-Intensive Applications. volume 0, 352–258. IEEE Computer Society, (2002).

[9] H. Sato, S. Matsuoka, T. Endo, and N. Maruyama. Access-Pattern and Bandwidth Aware File Replication Algorithm in a Grid Environment. In 'GRID', 250–257. IEEE, (2008).

[10] S. Shankar and D. J. DeWitt. Data Driven Workflow Planning in Cluster Management Systems. In 'HPDC '07', 127–136, New York, NY, USA, (2007). ACM.

[11] H. Simonis. *Constraint applications in networks.* In 'Handbook of Constraint Programming', F. Rossi, P. van Beek, and T. Walsh, (eds.), Elsevier (2006), chapter 25, 875–903.

[12] D. Teaff, D. Watson, and B. Coyne. The Architecture of the High Performance Storage System (HPSS). In 'Proceedings of the Goddard Conference on Mass Storage and Technologies', 28–30, (1995).

[13] M. Zerola, R. Barták, J. Lauret, and M. Šumbera. Efficient Multi-site Data Movement in Distributed Environment. In 'Proceedings of the 10^{th} IEEE/ACM International Conference on Grid Computing (GRID)', 'to appear'. IEEE, (2009).

[14] M. Zerola, R. Barták, J. Lauret, and M. Šumbera. Planning Heuristics for Efficient Data Movement on the Grid. In 'Proceedings of the 4^{th} Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA)', 768–771, (2009).

[15] M. Zerola, R. Barták, J. Lauret, and M. Šumbera. Using Constraint Programming to Plan Efficient Data Movement on the Grid. In 'Proceedings of the 21^{st} International Conference on Tools with Artificial Intelligence (ICTAI)', 'to appear'. IEEE, (2009).