

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra matematiky

DIPLOMOVÁ PRÁCE

Plzeň 2015

Barbora Jůzová

Prohlášení

Předkládám tímto k posouzení a obhajobě diplomovou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem diplomovou práci vypracovala samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

Plzeň 20. května 2015

.....

podpis

Anotace

Tato práce se zabývá vybranými problémy z oblasti numerického řešení parciálních diferenciálních rovnic a jejich aplikací na zpracování obrazu. Hlavní náplní je odvození některých numerických schémat, jejich implementace a porovnání výsledků s existujícími přístupy jiných technických oborů.

Klíčová slova: *Parciální diferenciální rovnice, numerické metody, metoda typu upwind, zpracování obrazu, filtrace, průměrování, gradientní operátory, Cannyho detektor, Laplaceův operátor, Sobelův filtr, šum, rozmazání, optimalizační úloha s totální variací, regularizace.*

Abstract

This work deals with selected problems from the area of numerical solution of partial differential equations and their applications to image processing. The main focus is to derive some numerical schemes, their implementation and results comparison with existing approaches of other technical disciplines.

Key words: *Partial differential equations, numerical methods, upwind method, image processing, filtering, averaging, gradient operators, Canny edge detector, Laplace operator, Sobel operator, noise, blurring, variation optimization problem, regularization.*

Obsah

1	Úvod	1
2	Základní pojmy	2
2.1	Barvy	2
2.1.1	Světlo	2
2.1.2	Mísení a reprezentace barev	2
2.2	Digitální obraz	3
2.3	Komplikace při zpracování obrazu	4
3	Parciální diferenciální rovnice	6
3.1	Definice	6
3.2	Příklady PDR	7
3.2.1	PDR prvního řádu - transportní rovnice	7
3.2.2	PDR druhého řádu	7
3.3	Numerické metody	8
3.3.1	Metoda typu upwind	9
3.3.2	Další metody pro lineární advekční rovnici	9
3.3.3	Využití centrálních diferencí	10
3.3.4	Řád metody	10
3.3.5	Level set metoda	11
3.3.6	Hartenova věta	11
4	Zpracování digitalizovaného obrazu - klasický přístup	13
4.1	Předzpracování	13
4.1.1	Bodové jasové transformace	14
4.1.2	Geometrické transformace	14
4.1.3	Frekvenční analýza	14
4.1.4	Lokální operace	14
5	Metody založené na PDR	18
5.1	Hyperbolická rovnice	18
5.1.1	Rozšíření hyperbolické rovnice do 2D	19
5.2	Parabolická rovnice	21
5.2.1	Rozšíření parabolické rovnice do 2D	21
5.3	Nelineární parabolická PDR	22
5.3.1	Rozšíření nelineární parabolické rovnice do 2D	25
6	Numerické experimenty	27
6.1	Rozmazání spojené se změnou souřadnic/pozice	27
6.2	Rozmazání bez pohybu obrázku	31
6.3	Odstranění šumu se zachováním hran v 1D	33
6.4	Odstranění šumu se zachováním hran v 2D	34
6.4.1	Odstranění šumu a detekce hran - jiné přístupy	41

6.4.2 Pozorování	46
7 Závěr	47
A Seznam příložených souborů	50

Seznam obrázků

2.1	Spektrum elektromagnetického záření	2
2.2	RGB a CMYK model	3
2.3	HSV model	3
2.4	Sousednost	4
4.1	Histogram	14
4.2	Základní směry	16
6.1	Rotační vektorové pole	27
6.2	Hyperbolická rovnice - posunutí	28
6.3	Hyperbolická rovnice - posunutí; vypočteno pomocí numerických toků	28
6.4	Hyperbolická rovnice - rotace válce	29
6.5	Hyperbolická rovnice - rotace kužele	29
6.6	Hyperbolická rovnice - rotace středově nesouměrného tělesa (seříznutého válce)	30
6.7	Hyperbolická rovnice - šedotónový obrázek	30
6.8	Hyperbolická rovnice - barevný obrázek	30
6.9	Parabolická rovnice - válec; CFL 1/2	31
6.10	Parabolická rovnice - kužel; CFL 1/2	31
6.11	Parabolická rovnice - středově nesouměrné těleso (seříznutý válec); CFL 1/2	32
6.12	Parabolická rovnice - válec; CFL 1/6	32
6.13	Parabolická rovnice - kužel; CFL 1/6	32
6.14	Parabolická rovnice - středově nesouměrné těleso (seříznutý válec); CFL 1/6	32
6.15	Parabolická rovnice - válec; nekonstantní parametry a, b	33
6.16	Parabolická rovnice - kužel; nekonstantní parametry a, b	33
6.17	Parabolická rovnice - středově nesouměrné těleso (seříznutý válec); nekonstantní parametry a, b	33
6.18	Odstranění šumu 1D	34
6.19	Ukázky šumu	35
6.20	Srovnání výstupů programu pro Gaussovský bílý šum - Lena	36
6.21	Srovnání výstupů programu pro Gaussovský šum - Lena	37
6.22	Srovnání výstupů programu pro šum sůl a pepř - Lena	38
6.23	Srovnání výstupů programu pro Gaussovský šum - Mona Lisa	39
6.24	Srovnání výstupů programu pro Gaussovský šum - kameraman	40
6.25	Volba okrajové podmínky 1	40
6.26	Volba okrajové podmínky 2	41
6.27	Ukázky průměrování	42
6.28	Ukázky průměrování	43
6.29	Frekvenční analýza	44
6.30	Ukázky detekce hran	45

Kapitola 1

Úvod

Zpracování digitalizovaného obrazu se používá v mnohých oblastech života, příkladem praktického využití je automatická detekce a rozpoznávání objektů. Problémem při takovém zpracování je přítomnost šumu, který se dostane do obrazu jak z důvodu technických nedostatků snímacího elementu (fotoaparátu), tak z důvodu nedostatečného osvětlení. Vědci z různých technických oborů hledají cesty jak odstranit šum, aniž by došlo ke ztrátě důležitých informací z počátečních dat.

Existuje více přístupů k problematice zpracování obrazu. Jedním z přístupů je využití numerických metod pro řešení parciálních diferenciálních rovnic, které jsou hlavní náplní této práce. Často se můžeme také setkat s redukcí šumu pomocí filtračních masek, což použijeme pro srovnání s numerickými metodami. Jedná se o postup používaný běžně především kybernetiky a informatiky. Z dalších metod jmenujme například iterační metody s regularizací.

V této diplomové práci se nejprve seznámíme se základními pojmy z oblasti počítačové grafiky (kapitola 2), abychom mohli zavést jednotný popis obrázků. V následujících kapitolách si ukážeme numerická schémata pro vybrané lineární i nelineární evoluční parciální diferenciální rovnice (kapitoly 3 a 5) a také několik nejběžnějších filtračních masek pro odstraňování šumu a detekci hran (kapitola 4). V kapitole 6 provedeme srovnání zmíněných přístupů z hlediska funkčnosti a efektivity.

Kapitola 2

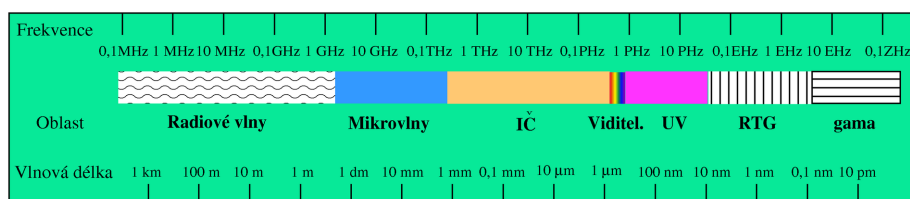
Základní pojmy

2.1 Barvy

2.1.1 Světlo

Světlem nazýváme elektromagnetické záření, což je příčné postupné vlnění elektrického a magnetického pole, o frekvenci $f = 3,9 \cdot 10^{14} - 7,9 \cdot 10^{14} \text{ Hz}$. Pro šíření světla ve vakuu rychlostí $c = 299\,792\,458 \text{ ms}^{-1}$ to odpovídá vlnové délce $\lambda = 390 - 790 \text{ nm}$ (platí vztah $c = f\lambda$). Kratší vlnovou délku má kosmické záření, gama paprsky, rentgenové záření a ultrafialové (UV) záření, které je pro člověka neviditelné, ale někteří živočichové, například plazi a ptáci, je dokáží vnímat. UV světlo zvyšuje lidem pigmentaci pokožky, ale může vést i k poruše metabolismu a vzniku rakoviny. Mezi záření s delší vlnovou délkou než světlo řadíme infračervené záření (IR), které vnímáme jako teplo, a rádiové vlny.

Nyní už se budeme zabývat pouze vlnovými délkami odpovídajícími viditelnému světlu. Různé frekvence světla vidíme jako barvy. Barva s nejnižší vlnovou délkou je fialová, s rostoucí vlnovou délkou přechází přes modrou, zelenou a žlutou k červené barvě, která má nejdelší vlnovou délku, tedy nejnižší frekvenci. Všechny takovéto barvy, odpovídající vlnové délce $\lambda = 390 - 790 \text{ nm}$, označujeme jako spektrální. Spektrum elektromagnetického záření je znázorněno na obr. 2.1.



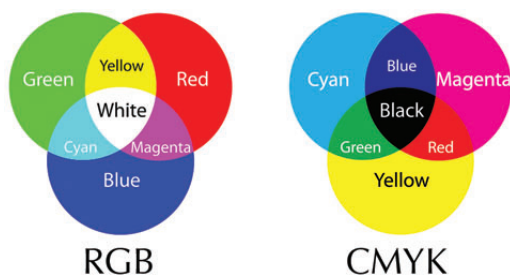
Obr. 2.1: Spektrum elektromagnetického záření (převzato z [1]).

2.1.2 Mísení a reprezentace barev

V lidském oku jsou receptory citlivé na tři základní barvy - červenou, zelenou a modrou, další odstíny vznikají skládáním těchto základních barev. Takové mísení nazýváme součtové neboli aditivní. Podle prvních písmen anglických názvů základních barev (**R**-red-červená, **G**-green-zelená, **B**-blue-modrá) je tento model nazván RGB, některé počítačové programy pracují s opačným pořadím barev (BGR). Bílou barvu získáme, jsou-li všechny složky maximální. Pokud jsou všechny složky nulové, výsledná barva je černá. To lze snadno vyzkoušet v počítačových programech typu Malování a k podobným výsledkům bychom měli dospět i při mísení malířských (temperových)

barev na paletě. Máme-li všechny složky zastoupeny stejnou měrou, nacházíme se na šedotónové škále mezi bílou a černou.

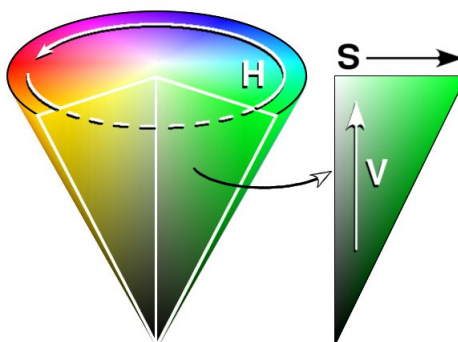
Druhým typem je rozdílové, neboli subtraktivní mísení, kdy se z dopadajícího světla odečítají určité barevné složky. Narozdíl od aditivního mísení, kdy maximální zastoupení všech složek vytvoří bílou barvu, v tomto případě vznikne barva černá. Bílou barvu získáme, jsou-li všechny složky nulové. Takovému mísení odpovídá CMY model (**C**-cyan-tyrkysová, **M**-magenta-fialová, **Y**-yellow-žlutá) využitý v tiskárnách. Někdy je doplněn barvou **K**-black-černá, která je „černější“ než maximální C+M+Y a také zabraňuje zbytečné spotřebě barevného pigmentu. RGB a CMYK model si můžete prohlédnout na obrázku 2.2.



Obr. 2.2: Srovnání RGB a CMYK modelu - mísení barev (převzato z [2]).

Další barevné odstíny můžeme získat přidáním černého (tzv. Shade = stín, který snižuje jas) nebo bílého (Tint = odstín, snižuje sytost) pigmentu do čistého pigmentu. Sytá barva patří mezi spektrální barvy a ztrácí sytost přidáváním bílé. Jas (nebo intenzita jasu) udává množství barevné složky. Hodnota jasu se pohybuje v intervalu $\langle 0, 255 \rangle$ pro každou ze tří složek (R,G,B), někdy jsou hodnoty jasu uváděny v procentech (rozsah $\langle 0, 1 \rangle$ u každé složky).

Pro některé obory a úkony (například segmentace - výběr oblastí se zadanými parametry) je výhodnější IHS model (**I**-intensity-intenzita R+G+B, **H**-hue-průměrná vlnová délka, **S**-saturation-sytost), nebo HSV model (**H**-hue-průměrná vlnová délka, **S**-saturation-sytost, **V**-value-hodnota jasu; obr. 2.3), které poskytují i jiná data, než RGB model.

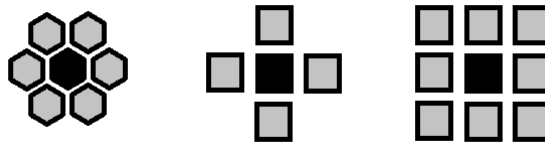


Obr. 2.3: HSV model (převzato z [3]).

2.2 Digitální obraz

Obraz na monitoru počítače si můžeme představit jako mozaiku z miniaturních jednobarevných dlaždic. Dlaždice mohou být trojúhelníkové, čtvercové nebo hexagonální - každý z těchto

mnohoúhelníků má své výhody i nevýhody při různém použití. Například pravidelný šestiúhelník nabízí stejnou vzdálenost středů sousedních dlaždic (ozn. R), zatímco ve čtvercové síti jsou středy některých sousedních dlaždic vzdáleny o R , jiné (úhlopříčné) o $R\sqrt{2}$. S tím souvisí i pojem okolí bodu. Šestiúhelníková dlaždice sousedí se šesti dalšími, se kterými má společnou jednu ze svých hran. Používáme-li čtvercovou síť, nejčastěji volíme *4-sousednost* (dlaždice, které mají se čtvercem právě jednu společnou hranu) nebo *8-sousednost* (uvažujeme i dlaždice, které mají se čtvercem jeden společný bod). Pokládáme za samozřejmé, že dlaždice se navzájem nepřekrývají a zároveň vyplňují celý prostor obrazu. Takové dlaždice nazýváme *pixely* (z angl. picture element = obrazový prvek). Pokud nepracujeme v rovině, ale v trojrozměrném prostoru, jednotlivé dlaždice nazýváme *voxely* (z angl. volumetric element = objemový prvek), které si můžeme představit jako krychlové dílky ve stavebnici. Různé typy sousednosti (pro 6, 4 a 8 sousedních dlaždic) jsou vyobrazeny na obr. 2.4.



Obr. 2.4: Sousednost; vlevo 6-okolí, uprostřed 4-okolí, vpravo 8-okolí.

S touto základní představou můžeme přejít k matematickému popisu obrazu. Dvourozměrný obraz máme rozdělený na $N \times M$ čtvercových dlaždic - pixelů a každý pixel je popsán třemi přirozenými čísly z intervalu $\langle 0, 255 \rangle$ (někdy se můžeme setkat s normovanými hodnotami z intervalu $\langle 0, 1 \rangle$), které odpovídají hodnotám jasu červené, zelené a modré složky. Tato stupnice byla zvolena kvůli reprezentaci čísel v počítači. V následující tabulce 2.1 jsou uvedeny příklady vybraných barev.

$[0, 0, 0]$	černá
$[255, 255, 255]$	bílá
$[255, 0, 0]$	červená
$[0, 255, 0]$	zelená
$[0, 0, 255]$	modrá
$[k, k, k], k \in \mathbb{N}, k \in (0, 255)$	odstíny šedé

Tab. 2.1: Základní barvy.

Abychom mohli s obrázkem dále pracovat, uložíme tyto kombinace čísel do matice $Im(m, n)$, kde pozice (m, n) odpovídá pořadí pixelu ve sloupci, resp. řádku čtvercové sítě obrázku. Protože je každý odstín barvy reprezentován třemi složkami, vytvoříme tři matice Im_R, Im_G, Im_B odpovídající základním barvám - červené, zelené a modré. Pracujeme-li se šedotónovým obrázkem, zachováme pouze jednu matici ze tří výše uvedených, protože u barev na šedotónové stupnici si jsou složky R, G a B rovny (viz tabulka). Takový obrázek pak lze vykreslit i jako trojrozměrný graf, kde souřadnice x a y odpovídají poloze pixelu v obrázku a souřadnice z udává hodnotu jasu.

2.3 Komplikace při zpracování obrazu

Trojrozměrné vidění člověku umožňuje skutečnost, že se na svět díváme dvěma očima, které jsou umístěny vedle sebe. Snímek zachycený fotoaparátlem ale třetí rozměr ztrácí, protože prostor před objektivem promítáme na dvourozměrnou plochu. Tím přicházíme o informaci o vzdálenosti jednotlivých předmětů a také nedokážeme určit, jestli pro nás nějaký objekt nezůstal skrytý za

jiným předmětem. Je to stejné, jako když zavřete jedno oko, pozorujete svůj ukazováček, prstem druhé ruky se k němu ze strany přibližujete a snažíte se ho dotknout. Chceme-li v počítači zachovat trojrozměrné vnímání, musíme scénu snímat dvěma fotoaparáty nebo videokamerami ze dvou různých míst. Někdy se do popisu obrázku přidává číslo udávající vzdálenost jednotlivých předmětů (pixelů) od pozorovatele (fotoaparátu). Vzhledem k diskrétnímu popisu také dochází k deformaci a zkreslení oblouků a hran svírajících s osami jiný úhel než celočíselné násobky 45° .

Kapitola 3

Parciální diferenciální rovnice

V předchozí kapitole jsme se seznámili se základními pojmy jako jsou barvy, pixel a sousednost. Protože se tato práce zabývá numerickými metodami pro řešení parciálních diferenciálních rovnic, věnujme se nyní základům tohoto matematického odvětví (dále popsané rovnice a metody jsou převzaty z [4] a [5]).

Obyčejné diferenciální rovnice (značíme ODR) obsahují neznámou funkci jedné proměnné a její derivace. Jejich zobecněním vzniknou parciální diferenciální rovnice (dále značené jako PDR) obsahující jako neznámé funkce více proměnných a jejich parciální derivace. PDR řešíme analyticky nebo numericky - existuje řada numerických metod, jmenujme například metodu konečných objemů, metodu konečných diferencí nebo metodu konečných prvků, vybrané metody pro některé rovnice budou popsány dále. PDR používáme pro formulaci řady problémů a k popisu široké škály jevů a dějů, uvedme například zvuk, teplo, elektrostatika, elektrodynamika, proudění tekutin a pružnost, také mohou sloužit k základním úpravám při zpracování obrazu.

3.1 Definice

Nejdříve definujeme obecnou PDR a některé speciální případy. Buď Ω otevřená podmnožina \mathbb{R}^n a buď dáno zobrazení

$$F : \Omega \times \mathbb{R} \times \mathbb{R}^n \times \dots \times \mathbb{R}^{n^{k-1}} \times \mathbb{R}^{n^k} \rightarrow \mathbb{R}. \quad (3.1)$$

Pak výraz

$$F(\mathbf{x}, q(\mathbf{x}), Dq(\mathbf{x}), D^2q(\mathbf{x}), \dots, D^kq(\mathbf{x})) = 0, \quad (3.2)$$

kde

$$\begin{aligned} Dq(\mathbf{x}) &= \left(\frac{\partial q}{\partial x_1}, \dots, \frac{\partial q}{\partial x_n} \right), \\ D^2q(\mathbf{x}) &= \left(\frac{\partial^2 q}{\partial x_1^2}, \frac{\partial^2 q}{\partial x_1 \partial x_2}, \dots, \frac{\partial^2 q}{\partial x_n^2} \right), \\ &\vdots \\ D^k &= \left(\frac{\partial^k q}{\partial x_1^{\alpha_1} \dots \partial x_k^{\alpha_k}} \mid [\alpha_1, \dots, \alpha_k] \geq 0; |\alpha| = \sum_{i=1}^n k_i = k \right), \end{aligned}$$

nazveme *parciální diferenciální rovnici k-tého řádu*. Funkce

$$q : \Omega \rightarrow \mathbb{R}$$

je neznámá. Symbol α se nazývá *multiindex*.

PDR se nazývá *lineární*, pokud ji lze zapsat ve tvaru

$$\sum_{|\alpha| \leq k} a_\alpha(\mathbf{x}) D^\alpha q(\mathbf{x}) = f(\mathbf{x}),$$

kde $f = f(x)$ je zadaná funkce. Je-li $f \equiv 0$, pak rovnici nazveme *homogenní lineární PDR*.

3.2 Příklady PDR

V literatuře se setkáváme s více různými náhledy na PDR. Uvažujme funkci $q = q(\mathbf{x})$. Rovnici značíme jako *lineární*, pokud je vztah mezi q a jejími derivacemi lineární, tedy pokud rovnice neobsahuje součiny typu qq_x, qq_{xy} ; v opačném případě hovoříme o nelineární rovnici. Nejvyšší řád derivace hledané funkce, který se v rovnici vyskytuje, určuje *řád* rovnice. Rozlišujeme *homogenní* a *nehomogenní rovnice* podle toho, jestli pravá strana je nebo není nulová. V *evolučních rovnicích* je čas t jednou z nezávisle proměnných, ve *stacionárních rovnicích* vystupují pouze prostorově nezávislé proměnné. V následujících odstavcích uvedeme několik základních typů PDR.

3.2.1 PDR prvního řádu - transportní rovnice

Bud $q : \mathbb{R} \times (0, \infty) \rightarrow \mathbb{R}$ funkce dvou proměnných x a t . Uvažujme rovnici

$$\frac{\partial q}{\partial t}(x, t) + a \frac{\partial q}{\partial x}(x, t) = 0 \quad (3.3)$$

na $\mathbb{R} \times (0, \infty)$, jednodušeji zapsáno

$$q_t + a q_x = 0 \quad (3.4)$$

na $\mathbb{R} \times (0, \infty)$, $a \in \mathbb{R}$. Konstantu a lze interpretovat jako rychlost proudění. Řešení této rovnice má tvar

$$q(x, t) = F(x - at), \quad (3.5)$$

kde F je libovolná diferencovatelná funkce. Pro $a > 0$ se takové řešení nazývá *pravá postupná vlna*, protože s rostoucím časem se „profil funkce“ pohybuje nezměněn doprava rychlostí a . Pokud je parametr a záporný, řešení nazveme *levou postupnou vlnou*, což odpovídá tomu, že proudící tekutina proudí doleva rychlostí o velikosti $|a|$.

Pro více prostorových proměnných můžeme transportní rovnici napsat ve tvaru

$$q_t + a q_x + b q_y = 0, \quad (3.6)$$

kde $a, b \in \mathbb{R}$.

Tuto lineární *advektivní* rovnici (výraz advekce se používá pro přenos látky prouděním) s nulovou pravou stranou a konstantními koeficienty můžeme modifikovat tím, že koeficienty a a b budou prostorově proměnné.

3.2.2 PDR druhého řádu

Rovnici napíšeme ve tvaru

$$a q_{xx} + b q_{xy} + c q_{yy} + d q_x + e q_y + f q = g(x, y), \quad (3.7)$$

kde $g = g(x, y)$ je zadaná funkce. Podle hodnoty diskriminantu $D = b^2 - 4ac$ rozlišujeme tři základní typy:

- $D > 0$ **hyperbolické**,
- $D = 0$ **parabolické**,
- $D < 0$ **eliptické**.

Nahradíme-li derivace proměnnými ($q_{xx} \rightarrow x^2$, $q_{xy} \rightarrow xy \dots$), můžeme si povšimnout podobnosti těchto rovnic s rovnicemi popisujícími kuželosečky.

Alternativním způsobem klasifikace je výpočet determinantu matice

$$\delta = \det \begin{bmatrix} a & b/2 \\ b/2 & c \end{bmatrix},$$

pak rozlišujeme eliptické ($\delta > 0$), parabolické ($\delta = 0$) a hyperbolické rovnice ($\delta < 0$).

Příklady rovnic pro jednotlivé typy:

- hyperbolické: vlnová rovnice $q_{tt} = c^2 q_{xx}$,
- parabolické: rovnice vedení tepla $q_t = \beta q_{xx}$, $\beta > 0$,
- eliptické: Poissonova rovnice $q_{xx} + q_{yy} = g(x, y)$.

U parabolických a hyperbolických rovnic je použito t místo y , protože se jedná o typické časově závislé rovnice.

Pro každý z těchto typů existují speciální metody a algoritmy řešení s ohledem na vlastnosti řešení. Rovnice také můžeme rozšířit do vyšších prostorových dimenzí. S rovnicemi se setkáváme při řešení úloh vedení tepla, chemické difúze a jiných oborů. Popisují průběh nějakého děje v čase. Vzhledem k výjimečnému charakteru časové proměnné se zadává počáteční podmínka a dále okrajové podmínky. Tyto tzv. počátečně-okrajové úlohy mají vůči t charakter počáteční, vzhledem k prostorovým proměnným jde o úlohy okrajové.

3.3 Numerické metody

V této sekci se seznámíme s numerickými metodami používanými při řešení PDR. Existují různé metody, nás budou zajímat hlavně metody konečných diferencí (FDM) a metody konečných objemů (FVM). Mezi ně se řadí i metoda *upwind*, ze které budeme vycházet při řešení problémů zabývajících se zpracováním obrazu.

V této diplomové práci se budeme zabývat hyperbolickou a parabolickou rovnicí. Aby byla odvozená numerická metoda stabilní, musíme splnit *CFL podmínku*

$$|a| \frac{\Delta t}{\Delta x} \leq 1 \quad (3.8)$$

v případě hyperbolické rovnice a

$$|a| \frac{\Delta t}{\Delta x^2} \leq 1 \quad (3.9)$$

pro parabolickou rovnicí, kde Δt je velikost časového kroku a Δx je velikost prostorového kroku diskretizace (podrobněji v následujících odstavcích). CFL podmínka u parabolické rovnice je velmi přísná, proto častěji volíme implicitní numerické schéma (pro výpočet nové časové vrstvy $n + 1$ potřebujeme hodnoty ze stejné časové vrstvy; řešíme většinou maticově soustavu rovnic). Naopak u hyperbolické rovnice si vystačíme s explicitní metodou (hodnoty v časové vrstvě označené $n + 1$ počítáme ze známých hodnot n -té časové vrstvy). V kapitole 5 se budeme zabývat jen vybranými metodami a konkrétními rovnicemi.

Odvodíme zde numerické schéma pro lineární advekční rovnici s nulovou pravou stranou v jedné prostorové dimenzi, u dalších rovnic budeme odvozovat schémata analogicky a uvedeme pouze výpočetní tvar metody.

Uvažujme počátečně-okrajovou úlohu

$$q_t + a q_x = 0, \quad a \in \mathbb{R} \quad (3.10)$$

$$x \in (L_1, L_2), \quad t \in (0, T)$$

$$q(L_1, t) = q_{L_1}, \quad t \in (0, T)$$

$$\begin{aligned} q(L_2, t) &= q_{L_2}, \quad t \in (0, T) \\ q(x, 0) &= q_0(x), \quad x \in (L_1, L_2). \end{aligned}$$

Stačí nám jedna okrajová podmínka, kterou vybíráme podle znaménka konstantního parametru a . V následujícím odstavci odvodíme metodu typu upwind pro řešení této úlohy.

Zavedeme síť

$$\begin{aligned} x_j &= j\Delta x, \quad \Delta x > 0, \quad j \in \{1, 2, \dots, M\}, \\ t_n &= n\Delta t, \quad \Delta t > 0, \quad n \in \{0, 1, \dots, N\}. \end{aligned}$$

Hodnotu přesného řešení v bodě (x_j, t_n) , tj. $q_j^n = q(x_j, t_n)$, aproximujeme hodnotou Q_j^n .

3.3.1 Metoda typu upwind

Nejprve odvodíme vztah pro výpočet Q_j^{n+1} pro kladné a . Převědeme prostorovou derivaci na pravou stranu

$$q_t = -aq_x.$$

Nyní použijeme metodu konečných diferencí, tzn. nahradíme derivace poměrnými diferencemi

$$\frac{Q_j^{n+1} - Q_j^n}{\Delta t} = -a \left(\frac{Q_j^n - Q_{j-1}^n}{\Delta x} \right).$$

Přenásobíme rovnost časovým krokem Δt a upravíme pravou stranu

$$Q_j^{n+1} = Q_j^n - \frac{a\Delta t}{\Delta x} (Q_j^n - Q_{j-1}^n). \quad (3.11)$$

Upravíme vztah pro $a < 0$ - v každém cyklu použijeme pro výpočet nové hodnoty jiné hodnoty síťové funkce Q_j^n (použijeme pravostranné diference)

$$\begin{aligned} q_t &= -aq_x \\ \frac{Q_j^{n+1} - Q_j^n}{\Delta t} &= -a \left(\frac{Q_{j+1}^n - Q_j^n}{\Delta x} \right) \\ Q_j^{n+1} &= Q_j^n - \frac{a\Delta t}{\Delta x} (Q_{j+1}^n - Q_j^n). \end{aligned} \quad (3.12)$$

Spojením vztahů (3.11) a (3.12) získáme obecný tvar

$$Q_j^{n+1} = Q_j^n - \frac{\Delta t}{\Delta x} [a^+ (Q_j^n - Q_{j-1}^n) + a^- (Q_{j+1}^n - Q_j^n)], \quad (3.13)$$

kde $a^+ = \max\{a, 0\}$ a $a^- = \min\{a, 0\}$.

Tato metoda je konzistentní, prvního řádu a stabilní při splnění CFL podmínky 3.8.

3.3.2 Další metody pro lineární advekční rovnici

Pro srovnání ještě uvedeme schémata pro Laxovu-Friedrichsovu a Laxovu-Wendroffovu metodu (odvození např. v [5]). I tyto metody jsou konzistentní a stabilní při splnění podmínky $|a| \frac{\Delta t}{\Delta x} \leq 1$, Laxova-Friedrichsova metoda je, stejně jako metoda typu upwind, prvního řádu, Laxova-Wendroffova metoda je druhého řádu.

- Laxova-Friedrichsova metoda

$$Q_j^{n+1} = \frac{1}{2} (Q_{j+1}^n + Q_{j-1}^n) - \frac{a\Delta t}{2\Delta x} (Q_{j+1}^n - Q_{j-1}^n). \quad (3.14)$$

- Laxova-Wendroffova metoda

$$Q_j^{n+1} = Q_j^n - \frac{a\Delta t}{2\Delta x} (Q_{j+1}^n - Q_{j-1}^n) + \frac{a^2 (\Delta t)^2}{2(\Delta x)^2} (Q_{j+1}^n - 2Q_j^n + Q_{j-1}^n). \quad (3.15)$$

3.3.3 Využití centrálních diferencí

Pro numerické řešení parabolické rovnice

$$q_t + aq_{xx} = 0 \quad (3.16)$$

se spojité prostorové derivace obvykle nahrazují centrálními diferencemi. Explicitní Eulerova metoda pak má tvar

$$Q_j^{n+1} = Q_j^n - \frac{a\Delta t}{\Delta x^2} (Q_{j+1}^n - 2Q_j^n + Q_{j-1}^n). \quad (3.17)$$

3.3.4 Řád metody

Často se při numerických experimentech měří řád metody, proto zde uvedeme několik variant, jak určit řád metody. Způsoby si ukážeme například pro lineární advekční rovnici s nulovou pravou stranou

$$q_t + aq_x = 0. \quad (3.18)$$

U některých rovnic jsme schopni stanovit přesné řešení stacionárního problému. Ustálené řešení znamená, že časová derivace je rovna nule, tedy platí

$$aq_x = 0. \quad (3.19)$$

Konstanta a derivaci nijak neovlivní. Z toho vidíme, že q je konstantní. Z okrajové podmínky $q(0) = p$ plyne, že $q(x) = p$, kde p je okrajová hodnota. Řád budeme určovat pro stacionární řešení, numerickou metodu pro evoluční úlohu chápeme jako iterační metodu.

Určíme reziduuum a chybu. Reziduuum je rozdíl hodnoty z aktuálního a minulého kroku

$$r(n) = \|Q^{n+1} - Q^n\|, \quad (3.20)$$

můžeme zvolit například následující normu

$$\|Q^{n+1} - Q^n\| = \sqrt{\sum_{j=1}^M (Q_j^{n+1} - Q_j^n)^2}. \quad (3.21)$$

Chyba je určena jako rozdíl přesného a numerického řešení

$$e(n) = \|Q^n - q^*\|, \quad (3.22)$$

kde q^* je přesné řešení, např.

$$\|Q^n - q^*\| = \Delta x \sum_{j=1}^M |Q_j^n - q^*|. \quad (3.23)$$

Nyní popíšeme způsob, jak stanovit numerický řád metody při znalosti přesného řešení stacionárního problému. Řád vypočítáme jako podíl chyb $e(n)$ a velikosti kroku Δx pro zvolené a poloviční Δx , chybu určíme jako rozdíl přesné a vypočítané hodnoty Q v posledním počítaném kroku.

Vztah pro výpočet řádu jsme odvodili podle [6]. Necht $U(T, x)$ je přesné řešení a $V_h(T, x)$ je numerické řešení problému v čase T

$$u_t + f(u)_x = 0, \quad u(0, x) = u_0(x), \quad x \in \mathbb{R}. \quad (3.24)$$

Numerické řešení je řádu r , pokud

$$V_h(T, x) - U(T, x) = Ch^r + o(h^r), \quad (3.25)$$

konstanta C není závislá na velikosti prostorového kroku h .

Pokud známe přesné řešení, stanovíme řád metody přímo ze vzorce

$$\text{order} = \log_2 \frac{e_{\Delta x}}{e_{\Delta x/2}}.$$

Předchozí vztah lze upravit do tvaru

$$\text{order} = \frac{\log \frac{e_{\Delta x}}{e_{\Delta x/2}}}{\log \frac{\Delta x}{\Delta x/2}}. \quad (3.26)$$

Pro složitější úlohu, kdy neznáme přesné řešení, odhadneme řád r následujícím způsobem. K výpočtu řádu (založeno na Rungeho pravidlu; [6]) stačí mít 3 numerické výsledky s poměrně malými prostorovými kroky $h_1 = h$, $h_2 = \frac{h}{2}$ a $h_3 = \frac{h}{4}$. Pro každý krok vztahu ((3.25)) je splněn, proto

$$\delta V_i = |V_{h_i} - V_{h_{i+1}}| = |C| (h_i - h_{i+1}) + o(h^r), \quad i = 1, 2.$$

S přesností $o(h^r)$ získáme

$$\frac{\delta V_1}{\delta V_2} = \frac{h_1^r - h_2^r}{h_2^r - h_3^r} = \frac{1 - \left(\frac{1}{2}\right)^r}{\left(\frac{1}{2}\right)^r - \left(\frac{1}{4}\right)^r} = 2^r,$$

z čehož plyne

$$r = \log_2 \frac{\delta V_1}{\delta V_2}, \quad (3.27)$$

kde δV_1 a δV_2 jsou chyby pro h_1 a h_2 .

Pokud je na daném intervalu řešení hladké, numerický řád se rovná teoretickému řádu, jinak je řád menší.

3.3.5 Level set metoda

Level set metoda je numerická metoda pro segmentaci objektů v obraze na základě detekce hran ([7], [8]). Byla vyvinuta v 80. letech minuleho století americkými matematiky Stanleyem Osherem a Jamesem A. Sethianem. Můžeme se s ní setkat v řadě odvětví, jako jsou zpracování obrazu, počítačová grafika, výpočetní geometrie, optimalizace a dynamika tekutin.

Level set modely reprezentují křivku $C(t)$ jako nulovou vrstevnici implicitně zadané funkce ϕ , dimenze funkce ϕ je vždy o jedničku vyšší než dimenze pohybujícího se rozhraní. Křivku rozpínáme, resp. stahujeme tak, aby se zachytávala o hrany. Tuto evoluci popisujeme parciálními diferenciálními rovnicemi.

Tato metoda je často používána pro řešení nelineární parabolické rovnice s advekčním a difuzním (difuzí se označuje proces rozptylování se částic v prostoru) členem - využíváme toho především k detekci hran. Nevýhodou level set metody je vysoká výpočetní náročnost, protože kromě samotného výpočtu je třeba v každé iteraci stanovit nové hodnoty funkce ϕ pro všechny body obrazu.

3.3.6 Hartenova věta

Pro schémata v kapitole 5 budeme potřebovat znalost Hartenovy věty, nejprve ale zavedeme pojem *TVD vlastnost*.

Definice. Řekneme, že metoda má vlastnost *TVD* (*total variation diminishing*), platí-li

$$TV_{n+1}(U_{\Delta t}) \leq TV_n(U_{\Delta t}), \quad (3.28)$$

kde $TV_n(U_{\Delta t}) = \sum_{j=-\infty}^{\infty} |U_{j+1}^n - U_j^n|$.

Jelikož nejsou předmětem zkoumání jen hladká data, musíme ošetřit stabilitu i pro nehladké řešení. U parabolických rovnic, které řešení vyhlazují, si vystačíme s lineární stabilitou, ale v případě hyperbolických rovnic nám k ošetření stability poslouží například právě vlastnost TVD (podrobněji např. v [5]).

Věta. *Explicitní metoda konečných diferencí ve tvaru*

$$Q_j^{n+1} = Q_i^n - C_{n,j-\frac{1}{2}}^- (Q_j^n - Q_{j-1}^n) + C_{n,j+\frac{1}{2}}^+ (Q_{j+1}^n - Q_j^n) \quad (3.29)$$

je TVD za předpokladu, že koeficienty splňují podmínky

$$C_{n,j+\frac{1}{2}}^+ \geq 0,$$

$$C_{n,j-\frac{1}{2}}^- \geq 0,$$

$$C_{n,j+\frac{1}{2}}^+ + C_{n,j-\frac{1}{2}}^- \leq 1.$$

Důkaz věty zde nebudeme uvádět, je popsán např. v [9].

Kapitola 4

Zpracování digitalizovaného obrazu - klasický přístup

Teorie v této kapitole je převzatá z [10].

4.1 Předzpracování

Úlohy odstranění šumu a ostření hran, kterými se zabývá tato práce, řadíme do kategorie předzpracování obrazu. Do pokročilejších kategorií patří například popis, rozpoznání a klasifikace objektů. Heslo, které nejvíce vypovídá o této činnosti, zní: „Nejlepší předzpracování je žádné předzpracování,“ protože největší množství informací obsahuje původní obraz, úpravami informace ztrácíme, případně přidáváme takové, které do původního obrazu nepatří. Cílem předzpracování tedy bývá „zjednodušení“ obrázku a odstranění přebytečných informací, abychom si usnadnili pozdější operace. Vstupem předzpracování je obraz $f(i, j)$, výstupem bývá nejčastěji obraz $g(i, j)$.

Metody dělíme podle velikosti okolí bodu použitého při výpočtu:

- bodové jasové transformace,
- geometrické transformace,
- frekvenční analýza,
- lokální operace.

Než se podrobněji podíváme na výše uvedené metody, vysvětlíme nejprve pojem *histogram*. Histogram $H(p)$ je funkcí jasu a vyjadřuje četnost bodů s daným jasnem p (ukázka histogramu na obrázku 4.1). Definujeme *absolutní histogram*

$$H(p) = \sum_{i,j} h(i, j, p), \quad (4.1)$$

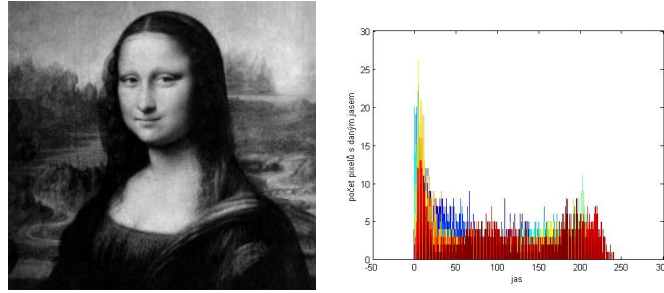
kde

$$h(i, j, p) = 1 \text{ pro } f(i, j) = p,$$

$$h(i, j, p) = 0 \text{ pro } f(i, j) \neq p.$$

Někdy se můžeme setkat s pojmem *relativní histogram*

$$H_R(p) = \frac{H(p)}{\sum_p H(p)} = \frac{H(p)}{ij}.$$



Obr. 4.1: Šedotónový obrázek s histogramem (obrázek převzat z [11], histogram vytvořen v programu MATLAB).

4.1.1 Bodové jasové transformace

Samotný název této kategorie transformací vypovídá o počtu pixelů vstupního obrázku, které potřebujeme k vypočtení hodnoty jasu jednoho pixelu výstupního obrázku - tedy jeden pixel. Řadíme sem jasové korekce - nasnímáme obraz se známými hodnotami, které porovnáme se získanými údaji; každý další snímání obrázek pak násobíme maticí opravných koeficientů $Opr(i, j)$ definovanou podílem správné a naměřené hodnoty pro každý pixel

$$Opr(i, j) = \frac{spr.hodnoty(i, j)}{nam.hodnoty(i, j)}. \quad (4.2)$$

Známějšími úpravami jsou jasové transformace, kdy je pro $g(i, j) = F(f(i, j))$ funkce F stejná pro všechny pixely. Mezi takovéto úpravy patří například zvýšení/snížení jasu a kontrastu nebo vytvoření negativu.

4.1.2 Geometrické transformace

Geometrickými transformacemi rozumíme rotaci, posun a zvětšení, ale také deformaci na základě tzv. vřícovacích bodů (toho se využívá například u družicových snímků). Protože takové operace většinou nezobrazí původní pixel přesně do nové souřadnice (uvažujeme souřadnice $i, j \in \mathbb{N}$), následuje určení hodnoty jasu v daném pixelu z hodnot v okolí - například volba nejbližšího souseda nebo bilineární či bikubická interpolace.

4.1.3 Frekvenční analýza

Využíváme skutečnosti, že vyšší frekvence představují šum. Pro výpočet používáme diskrétní Fourierovu transformaci

$$F(u, v) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f(n, m) e^{(-2\pi i(\frac{nu}{N} + \frac{mv}{M}))}. \quad (4.3)$$

4.1.4 Lokální operace

Hodnotu jasu pixelu ve výstupním obrázku vypočteme diskrétní *konvolucí* z hodnot jasu blízkého okolí pixelu ve vstupním obrázku

$$g(i, j) = \sum_{m=-1}^1 \sum_{n=-1}^1 f(i-m, j-n) h(n, m). \quad (4.4)$$

Výše uvedený vztah platí pro 8-okolí bodu, matici $h(m, n)$ nazveme maskou, v tomto případě má matice h rozměr $[3, 3]$; podobně vypadají i masky pro větší zvolené okolí, jen symetricky zvětšujeme matici h .

Vyhlazování - potlačení šumu

Myšlenkou vyhlazování je stanovit pro každý pixel průměr jasu v daném bodě a jeho okolí. Kromě aritmetického průměru se můžeme setkat i s různými variantami vážených průměrů:

- rovnoměrná maska

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix},$$

- zvýhodnění středového bodu

$$h = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix},$$

- zvýhodnění středového bodu a bodů v hlavních směrech, tato varianta je výhodnější pro strojové výpočty (bitový posun o 4 bity)

$$h = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix},$$

- znevýhodnění středového bodu, tato varianta je výhodnější pro strojové výpočty (bitový posun o 3 bity)

$$h = \frac{1}{8} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Pro všechny tyto masky platí, že součet hodnot v masce se rovná jedné. Nevýhodou je ztráta detailů, rozostření hran a také skutečnost, že liší-li se jeden pixel výrazně od ostatních, ovlivní znatelně jas v celém svém okolí. Tomu lze předejít například vybráním hodnoty, která je v okolí bodu nejčastěji zastoupena, problémem je tentokrát

- více hodnot jasu s maximálním zastoupením. například obsahuje-li okolí bodu (i, j) hodnoty

$$\begin{bmatrix} 10 & 93 & 245 \\ 10 & 93 & 245 \\ 10 & 93 & 245 \end{bmatrix},$$

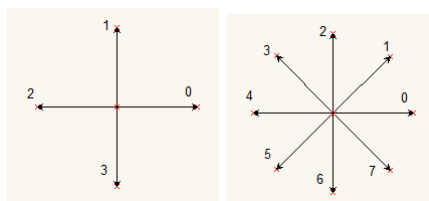
- všechny hodnoty jasu odlišné, například

$$\begin{bmatrix} 41 & 43 & 47 \\ 48 & 52 & 44 \\ 46 & 49 & 50 \end{bmatrix}.$$

Ještě lepší volbou je medián - zastoupené hodnoty seřadíme vzestupně a zvolíme tu prostřední. Takový filtr je vhodný pro redukci osamělých vychýlených hodnot, navíc nevznikají nové hodnoty jasu jako při průměrování; dochází ale k trhání rohů a porušování tenkých čar.

Gradientní operátory - detekce hran

V místech, kde se v digitálním obrazu prudce mění hodnoty jasu, očekáváme hranu. Zajímá nás, jak velký je rozdíl mezi sousedními jasy a také směr změny. Pokud hranu definujeme jako velkou změnu jasové funkce, bude v místě hrany velká hodnota derivace jasové funkce. Pro jednoduchost hledáme pouze hrany ve čtyřech, respektive osmi základních směrech (obr. 4.2).



Obr. 4.2: Základní směry.

Ve spojitém případě vyjádříme velikost gradientu výrazem

$$|\text{grad } g| = \sqrt{\left(\frac{\partial g}{\partial x}\right)^2 + \left(\frac{\partial g}{\partial y}\right)^2} \quad (4.5)$$

a směr gradientu vztahem

$$\varphi = \arctan\left(\frac{\frac{\partial g}{\partial y}}{\frac{\partial g}{\partial x}}\right). \quad (4.6)$$

V diskrétním případě aproximujeme derivace diferencemi

$$\Delta_x g(i, j) = g(i, j) - g(i, j - 1),$$

$$\Delta_y g(i, j) = g(i, j) - g(i - 1, j),$$

tedy

$$|\text{grad } g| = \sqrt{(\Delta_x g)^2 + (\Delta_y g)^2}, \quad (4.7)$$

$$\varphi = \arctan \frac{\Delta_y g}{\Delta_x g}. \quad (4.8)$$

Masky jsou vytvořeny tak, aby byl výsledek konvoluce na plochách blízky nule a v místě hrany co největší.

Uvedeme zde několik jednoduchých a často používaných operátorů:

- *Roberts* - nedokážeme určit směr, jen nespojitost jasové funkce; dnes už se moc nepoužívá, protože často detekuje šum jako hrany

$$g(i, j) = |f(i, j) - f(i + 1, j + 1)| + |f(i, j + 1) - f(i + 1, j)|,$$

$$h_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, h_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

- *Laplace* - udává pouze velikost hrany, nikoliv její směr; tato metoda je založená na druhé derivaci, můžeme vyjádřit diference

$$\Delta_x^2 f(i, j) = f(i, j + 1) + f(i, j - 1) - 2f(i, j),$$

$$\Delta_y^2 f(i, j) = f(i + 1, j) + f(i - 1, j) - 2f(i, j),$$

$$\nabla^2 f(i, j) = \Delta_x^2 f(i, j) + \Delta_y^2 f(i, j) = f(i, j + 1) + f(i, j - 1) + f(i + 1, j) + f(i - 1, j) - 4f(i, j)$$

a masku

$$h = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

- *Prewitt* - rozlišujeme osm masek pro osm základních směrů

$$h_0 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}, h_1 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix}, h_2 = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}, h_3 = \begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix},$$

$$h_4 = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, h_5 = \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, h_6 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, h_7 = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}.$$

Velikost gradientu určíme

$$|\text{grad } g| = \max_{k \in (0,7)} \{g * h_k\}, \quad (4.9)$$

a směr φ je k -násobkem úhlu 45° .

Uvedme ještě operátor velikosti 5×5 pro detekci hrany rovnoběžné s osou y , další směry bychom získali podobně jako v případě operátoru 3×3 ; zvětšujeme-li operátor, výpočet bude odolnější vůči šumu

$$h = \begin{bmatrix} -2 & -2 & -2 & -2 & -2 \\ -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \end{bmatrix}.$$

- *Sobel* - také pro tento operátor existuje osm variant pro jednotlivé směry:

$$h_0 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, h_1 = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}, h_2 = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, h_3 = \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix},$$

$$h_4 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, h_5 = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}, h_6 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, h_7 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}.$$

- *Kirsch* - opět používáme osm masek:

$$h_0 = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{bmatrix}, h_1 = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & -5 \\ 3 & -5 & -5 \end{bmatrix}, h_2 = \begin{bmatrix} 3 & 3 & -5 \\ 3 & 0 & -5 \\ 3 & 3 & -5 \end{bmatrix}, h_3 = \begin{bmatrix} 3 & -5 & -5 \\ 3 & 0 & -5 \\ 3 & 3 & 3 \end{bmatrix},$$

$$h_4 = \begin{bmatrix} -5 & -5 & -5 \\ 3 & 0 & 3 \\ 3 & 3 & 3 \end{bmatrix}, h_5 = \begin{bmatrix} -5 & -5 & 3 \\ -5 & 0 & 3 \\ 3 & 3 & 3 \end{bmatrix}, h_6 = \begin{bmatrix} -5 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & 3 & 3 \end{bmatrix}, h_7 = \begin{bmatrix} 3 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & -5 & 3 \end{bmatrix}.$$

Podrobnější popis a odvození gradientních operátorů naleznete např. v [12], [13] a [14].

Kapitola 5

Metody založené na PDR

V této kapitole odvodíme numerická schémata pro konkrétní parabolické a hyperbolické rovnice v jedné a dvou dimenzích a poté parabolickou rovnici rozšíříme o nelineární člen. Tato schémata jsou základem algoritmů pro zpracování obrazu.

5.1 Hyperbolická rovnice

Pro hyperbolickou rovnici v jedné dimenzi

$$q_t + aq_x = 0 \quad (5.1)$$

jsme v kapitole 3.3.1 odvodili explicitní numerickou upwind metodu ve tvaru

$$Q_j^{n+1} = Q_j^n - \frac{a\Delta t}{\Delta x} (Q_j^n - Q_{j-1}^n) \quad (5.2)$$

pro $a > 0$ a

$$Q_j^{n+1} = Q_j^n - \frac{a\Delta t}{\Delta x} (Q_{j+1}^n - Q_j^n) \quad (5.3)$$

pro $a < 0$ (rovnice (3.11) a (3.12)). Metodu upwind jsme zvolili pro její snadnou implementaci a také proto, že v porovnání s dalšími metodami ([15]) méně rozmazává (Laxova-Friedrichsova metoda) a neosciluje (Laxova-Wendroffova metoda). Můžeme se setkat se zápisem pomocí tokových funkcí ([5]), které pomáhají zachovat konzervativitu,

$$Q_j^{n+1} = Q_j^n - \frac{\Delta t}{\Delta x} [F_j^n - F_{j-1}^n], \quad (5.4)$$

kde

$$F_j^n = \frac{1}{2}a (Q_{j+1}^n + Q_j^n) - \frac{1}{2}|a| (Q_{j+1}^n - Q_j^n). \quad (5.5)$$

Dosadíme tokové funkce (5.5) přímo do zápisu metody (5.4) a upravíme na tvar

$$Q_j^{n+1} = Q_j^n - \frac{\Delta t}{\Delta x} \left(\frac{1}{2}a - \frac{1}{2}|a| \right) (Q_{j+1}^n - Q_j^n) - \frac{\Delta t}{\Delta x} \left(\frac{1}{2}a + \frac{1}{2}|a| \right) (Q_j^n - Q_{j-1}^n). \quad (5.6)$$

Označme

$$C_{n,j+\frac{1}{2}}^+ = \frac{\Delta t}{\Delta x} \left(\frac{1}{2}|a| - \frac{1}{2}a \right),$$

$$C_{n,j-\frac{1}{2}}^- = \frac{\Delta t}{\Delta x} \left(\frac{1}{2}|a| + \frac{1}{2}a \right).$$

Podle Hartenovy věty (3.3.6) lze metodu zapsat ve tvaru

$$Q_j^{n+1} = Q_i^n - C_{n,j-\frac{1}{2}}^- (Q_j^n - Q_{j-1}^n) + C_{n,j+\frac{1}{2}}^+ (Q_{j+1}^n - Q_j^n) \quad (5.7)$$

za podmínek

$$C_{n,j+\frac{1}{2}}^+ \geq 0,$$

$$C_{n,j-\frac{1}{2}}^- \geq 0$$

a

$$C_{n,j+\frac{1}{2}}^+ + C_{n,j-\frac{1}{2}}^- \leq 1.$$

Do této chvíle jsme uvažovali pouze explicitní numerická schémata - tedy hodnoty v nové časové vrstvě jsme počítali ze známých hodnot z předchozí vrstvy. Výhodou explicitní metody je rychlejší a jednodušší výpočet, ale volba velikosti časového kroku je omezena CFL podmínkou, která v tomto případě nabývá tvaru

$$|a| \frac{\Delta t}{\Delta x} + |b| \frac{\Delta t}{\Delta y} \leq 1. \quad (5.8)$$

Opačným případem, než explicitní metoda (5.2), je ryze implicitní metoda, například

$$Q_j^{n+1} + \frac{a\Delta t}{\Delta x} (Q_j^{n+1} - Q_{j-1}^{n+1}) = Q_j^n, \quad (5.9)$$

nebo kombinace obou rovnic (5.2) a (5.9)

$$Q_j^{n+1} + \frac{a\Delta t}{\Delta x} \eta (Q_j^{n+1} - Q_{j-1}^{n+1}) = Q_j^n - \frac{a\Delta t}{\Delta x} (1 - \eta) (Q_j^n - Q_{j-1}^n). \quad (5.10)$$

Dosazením hodnoty $\eta = 1$ získáme ryze implicitní metodu, pro $\eta = 0$ se metoda stává explicitní. Této vlastnosti se často využívá v různých výpočtech, kde se hodnota η mění a snadno se tak přepíná mezi implicitní a explicitní metodou. Výpočty s těmito metodami obvykle realizujeme v maticovém zápisu invertováním matice koeficientů na levé straně. Dle Hartenovy věty můžeme tuto metodu rozepsat stejným způsobem jako explicitní metodu, označíme-li

$$C_{n+1,j+\frac{1}{2}}^+ = \frac{\Delta t}{\Delta x} \left(\frac{1}{2}|a| - \frac{1}{2}a \right),$$

$$C_{n+1,j-\frac{1}{2}}^- = \frac{\Delta t}{\Delta x} \left(\frac{1}{2}|a| + \frac{1}{2}a \right),$$

pak

$$\begin{aligned} Q_j^{n+1} - \eta C_{n+1,j+\frac{1}{2}}^+ (Q_{j+1}^{n+1} - Q_j^{n+1}) + \eta C_{n+1,j-\frac{1}{2}}^- (Q_j^{n+1} - Q_{j-1}^{n+1}) &= \dots \\ \dots &= Q_i^n - (1 - \eta) C_{n,j-\frac{1}{2}}^- (Q_j^n - Q_{j-1}^n) + (1 - \eta) C_{n,j+\frac{1}{2}}^+ (Q_{j+1}^n - Q_j^n). \end{aligned} \quad (5.11)$$

5.1.1 Rozšíření hyperbolické rovnice do 2D

Rozšíříme nyní hyperbolickou rovnici do dvou prostorových dimenzí

$$q_t + aq_x + bq_y = 0. \quad (5.12)$$

Podobně jako v jedné dimenzi můžeme zapsat numerickou upwind metodu ve tvaru

$$Q_{i,j}^{n+1} = Q_{i,j}^n - \frac{a\Delta t}{\Delta x} (Q_{i,j}^n - Q_{i-1,j}^n) - \frac{b\Delta t}{\Delta x} (Q_{i,j}^n - Q_{i,j-1}^n) \quad (5.13)$$

pro $a > 0$, $b > 0$. Analogicky bychom zapsali schémata pro ostatní kombinace kladných/záporných konstant a , b . Prozatím jsme se vrátili k explicitní metodě.

Pomocí tokových funkcí můžeme souhrnně zapsat schéma

$$Q_{i,j}^{n+1} = Q_{i,j}^n - \frac{\Delta t}{\Delta x} [F_{i,j}^n - F_{i-1,j}^n] - \frac{\Delta t}{\Delta y} [C_{i,j}^n - C_{i,j-1}^n], \quad (5.14)$$

kde

$$F_{i,j}^n = \frac{1}{2}a (Q_{i+1,j}^n + Q_{i,j}^n) - \frac{1}{2}|a| (Q_{i+1,j}^n - Q_{i,j}^n), \quad (5.15)$$

$$F_{i-1,j}^n = \frac{1}{2}a (Q_{i,j}^n + Q_{i-1,j}^n) - \frac{1}{2}|a| (Q_{i,j}^n - Q_{i-1,j}^n), \quad (5.16)$$

$$G_{i,j}^n = \frac{1}{2}b (Q_{i,j+1}^n + Q_{i,j}^n) - \frac{1}{2}|b| (Q_{i,j+1}^n - Q_{i,j}^n), \quad (5.17)$$

$$G_{i-1,j}^n = \frac{1}{2}b (Q_{i,j}^n + Q_{i,j-1}^n) - \frac{1}{2}|b| (Q_{i,j}^n - Q_{i,j-1}^n). \quad (5.18)$$

Podle Hartenovy věty dosazením tokových funkcí ((5.15), (5.16), (5.17) a (5.18)) do schématu (5.14) a upravením výrazu přepíšeme metodu (nyní již uvažujeme přepínací schéma mezi explicitní a implicitní metodou) do podoby

$$\begin{aligned} & Q_{i,j}^{n+1} - \eta C_{n+1,i+\frac{1}{2},j}^+ (Q_{i+1,j}^{n+1} - Q_{i,j}^{n+1}) + \eta C_{n+1,i-\frac{1}{2},j}^- (Q_{i,j}^{n+1} - Q_{i-1,j}^{n+1}) - \dots \\ & \dots - \eta C_{n+1,i,j+\frac{1}{2}}^+ (Q_{i,j+1}^{n+1} - Q_{i,j}^{n+1}) + \eta C_{n+1,i,j-\frac{1}{2}}^- (Q_{i,j}^{n+1} - Q_{i,j-1}^{n+1}) = \dots \\ \dots = & Q_{i,j}^n - (1-\eta)C_{n,i-\frac{1}{2},j}^- (Q_{i,j}^n - Q_{i-1,j}^n) + (1-\eta)C_{n,i+\frac{1}{2},j}^+ (Q_{i+1,j}^n - Q_{i,j}^n) - \dots \\ & \dots - (1-\eta)C_{n,i,j-\frac{1}{2}}^- (Q_{i,j}^n - Q_{i,j-1}^n) + (1-\eta)C_{n,i,j+\frac{1}{2}}^+ (Q_{i,j+1}^n - Q_{i,j}^n), \end{aligned} \quad (5.19)$$

kde

$$C_{n,i+\frac{1}{2},j}^+ = \frac{\Delta t}{\Delta x} \left(\frac{1}{2}|a| - \frac{1}{2}a \right),$$

$$C_{n,i-\frac{1}{2},j}^- = \frac{\Delta t}{\Delta x} \left(\frac{1}{2}|a| + \frac{1}{2}a \right),$$

$$C_{n+1,i+\frac{1}{2},j}^+ = \frac{\Delta t}{\Delta x} \left(\frac{1}{2}|a| - \frac{1}{2}a \right),$$

$$C_{n+1,i-\frac{1}{2},j}^- = \frac{\Delta t}{\Delta x} \left(\frac{1}{2}|a| + \frac{1}{2}a \right),$$

$$C_{n,i,j+\frac{1}{2}}^+ = \frac{\Delta t}{\Delta y} \left(\frac{1}{2}|b| - \frac{1}{2}b \right),$$

$$C_{n,i,j-\frac{1}{2}}^- = \frac{\Delta t}{\Delta y} \left(\frac{1}{2}|b| + \frac{1}{2}b \right),$$

$$C_{n+1,i,j+\frac{1}{2}}^+ = \frac{\Delta t}{\Delta y} \left(\frac{1}{2}|b| - \frac{1}{2}b \right),$$

$$C_{n+1,i,j-\frac{1}{2}}^- = \frac{\Delta t}{\Delta y} \left(\frac{1}{2}|b| + \frac{1}{2}b \right).$$

U takovýchto metod často realizujeme tzv. linearizaci, kdy se hodnoty C_{n+1} nahrazují C_n .

5.2 Parabolická rovnice

Nyní se budeme věnovat parabolické rovnici

$$q_t + (aq_x)_x = 0, \quad (5.20)$$

pro začátek zůstaneme v jedné dimenzi. Konstantu a uvažujeme zápornou, rovnici můžeme přepsat na

$$q_t + aq_{xx} = 0. \quad (5.21)$$

Použijeme centrální diference a numerické schéma

$$Q_j^{n+1} = Q_j^n - a \frac{Q_{j+1}^n - 2Q_j^n + Q_{j-1}^n}{\Delta x^2} \quad (5.22)$$

přepíšeme pomocí tokových funkcí:

$$Q_j^{n+1} = Q_j^n - \frac{1}{\Delta x} \left[F_{j+\frac{1}{2}}^n - F_{j-\frac{1}{2}}^n \right], \quad (5.23)$$

kde

$$F_{j+\frac{1}{2}}^n = a \frac{Q_{j+1}^n - Q_j^n}{\Delta x}, \quad (5.24)$$

$$F_{j-\frac{1}{2}}^n = a \frac{Q_j^n - Q_{j-1}^n}{\Delta x}. \quad (5.25)$$

Také parabolickou rovnici můžeme řešit explicitně, jak je uvedeno výše, nebo implicitně.

5.2.1 Rozšíření parabolické rovnice do 2D

Ve dvou dimenzích má numerická metoda pro parabolickou rovnici

$$q_t + (aq_x)_x + (bq_y)_y = 0, \quad (5.26)$$

pro záporné konstantní parametry a a b

$$q_t + aq_{xx} + bq_{yy} = 0, \quad (5.27)$$

tvár

$$Q_{i,j}^{n+1} = Q_{i,j}^n - a \frac{Q_{i+1,j}^n - 2Q_{i,j}^n + Q_{i-1,j}^n}{\Delta x^2} - b \frac{Q_{i,j+1}^n - 2Q_{i,j}^n + Q_{i,j-1}^n}{\Delta y^2}. \quad (5.28)$$

Zápis s tokovými funkcemi vypadá podobně jako v jedné dimenzi, tedy

$$Q_{i,j}^{n+1} = Q_{i,j}^n - \frac{1}{\Delta x} \left[F_{i+\frac{1}{2},j}^n - F_{i-\frac{1}{2},j}^n \right] - \frac{1}{\Delta y} \left[G_{i,j+\frac{1}{2}}^n - G_{i,j-\frac{1}{2}}^n \right], \quad (5.29)$$

kde

$$F_{i+\frac{1}{2},j}^n = a \frac{Q_{i+1,j}^n - Q_{i,j}^n}{\Delta x}, \quad (5.30)$$

$$F_{i-\frac{1}{2},j}^n = a \frac{Q_{i,j}^n - Q_{i-1,j}^n}{\Delta x}, \quad (5.31)$$

$$G_{i,j+\frac{1}{2}}^n = b \frac{Q_{i,j+1}^n - Q_{i,j}^n}{\Delta y}, \quad (5.32)$$

$$G_{i,j-\frac{1}{2}}^n = b \frac{Q_{i,j}^n - Q_{i,j-1}^n}{\Delta y}. \quad (5.33)$$

CFL podmínka pro parabolickou rovnici ve dvou dimenzích omezuje volbu časového kroku vztahem

$$|a| \frac{\Delta t}{\Delta x^2} + |b| \frac{\Delta t}{\Delta y^2} \leq \frac{1}{2}. \quad (5.34)$$

Kvůli této přísné podmínce se parabolická rovnice častěji řeší implicitně.

5.3 Nelineární parabolická PDR

Do této chvíle jsme zde uváděli, popisovali a zkoumali pouze lineární parciální diferenciální rovnice. Jak bude patrné v následující kapitole, takové rovnice můžeme využít jen k jednoduchým úpravám obrázků, jmenujme například rozmazání (to může být jednou z cest, jak redukovat nežádoucí šum). Úkolem matematiků, kteří se zabývají zpracováním obrázků, je hledání takové rovnice a metody, která poškozený obrázek zbaví šumu aniž by ztratil ostrost, případně obnoví rozmazané hrany. Také se snaží zabránit nežádoucím efektům, jakým je například *staircase effect*, kdy se na obrázku vytváří nepravidelné plochy stejné barvy působící umělým dojmem.

Přístupme tedy k odvození aproximace nelineární parabolické rovnice vhodné pro tyto účely. Následující odvození metody je inspirováno článkem [16] a disertační prací [17]. Zkusme nyní na problém nahlížet z druhé strany. Označme $u_0(x, y)$ pixely čistého obrázku a $u(x, y)$ pixely poškozeného obrázku. Poškozením rozumíme

- šum, který popíšeme přidáním funkce $n(x, y)$ k čistému obrázku u_0 :

$$u = u_0 + n,$$

- rozmazání definované lineárním operátorem K :

$$u = K u_0,$$

- obojí výše jmenované, tedy rozmazaný obrázek se šumem:

$$u = K u_0 + n.$$

Zavedeme funkcionál R , který měří kvalitu obrázku u v tom smyslu, že menší hodnoty $R(u)$ odpovídají „lepší“ obrázkům; vycházíme ze statistik šumu a rozmazání a nějakých *a priori* znalostí o obrázku, jmenujme např. hladkost a existenci hran.

Řešíme minimalizační úlohu

$$\min_u R(u) \tag{5.35}$$

s vazbou

$$\|j * u - u_0\|_{\mathcal{L}^2}^2 = |\Omega| \sigma^2, \tag{5.36}$$

kde Ω představuje oblast obrázku. Tato podmínka plyne z $n = u_0 - j * u$ a $E\left(\int_{\Omega} n^2 dx\right) = |\Omega| \sigma^2$, kde $E(X)$ značí střední hodnotu náhodné proměnné X , pak

$$\|j * u - u_0\|_{\mathcal{L}^2}^2 = \int_{\Omega} (j * u - u_0)^2 dx \approx |\Omega| \sigma^2.$$

Funkcionál $R(u)$ může obsahovat například Laplacián ($R(u) = \|\Delta u\|_{\mathcal{L}^2}$) nebo gradient ($R(u) = \|\nabla u\|_{\mathcal{L}^2}$) funkce u . Jejich nevýhodou je, že nedovolují nespojitosti v řešení, proto při jejich použití nejsou hrany uspokojivě obnovovány.

Využijeme vlastností *TV-normy* jako regularizačního funkcionálu pro problém restaurování obrázku

$$TV(u) = \int_{\Omega} |\nabla u| dx = \int_{\Omega} \sqrt{u_x^2 + u_y^2} dx.$$

TV norma nepostihuje nespojitosti v u - to nám dovoluje obnovit hrany původního obrázku. Existují i další funkcionály s podobnými vlastnostmi, podrobněji v [16].

Přepíšeme úlohu restaurování obrázku na

$$\min_u \int_{\Omega} |\nabla u| dx \tag{5.37}$$

s vazbou

$$\frac{1}{2} \left(\int_{\Omega} (j * u - u_0)^2 dx - |\Omega| \sigma^2 \right) = 0. \quad (5.38)$$

Lagrangian této úlohy je

$$\int_{\Omega} |\nabla u| dx + \frac{\lambda}{2} \left(\int_{\Omega} (j * u - u_0)^2 dx - |\Omega| \sigma^2 \right)$$

a jeho *Eulerovy-Lagrangeovy rovnice s homogenními Neumannovými okrajovými podmínkami* pro u ($\frac{\partial u}{\partial n} = 0$) jsou

$$0 = -\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + \lambda (j * (j * u - u_0)), \quad (5.39)$$

$$0 = \frac{1}{2} \left(\int_{\Omega} (j * u - u_0)^2 dx - |\Omega| \sigma^2 \right). \quad (5.40)$$

Jsou známy techniky na řešení omezeného optimalizačního problému (5.37) využívající řešiče odpovídajícího problému bez vazeb s Eulerovou-Lagrangeovou rovnicí (5.39) pro pevné λ . Proto kvůli lepší přehlednosti budeme předpokládat, že Lagrangeův multiplikátor λ bude znám po celou dobu výpočtu.

Přepíšeme odpovídající problém bez vazby pro $\nu = \frac{1}{\lambda}$

$$\min_u \int_{\Omega} \left(\nu |\nabla u| + \frac{1}{2} (j * u - u_0)^2 \right) dx$$

a jeho Eulerovu-Lagrangeovu rovnici v běžnějším tvaru

$$-\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + \lambda j * (j * u - u_0) = 0 \quad (5.41)$$

jedná se o nelineární dekonvolutivní model. Lineární dekonvolutivní model

$$-\Delta u + \lambda j * (j * u - u_0) = 0 \quad (5.42)$$

vychází z Eulerovy-Lagrangeovy rovnice s normou $R(u) = \|\nabla u\|_{\mathcal{L}^2}$.

Abychom zabránili situaci, kdy bude jmenovatel nulový ($\nabla u = 0$), definujeme

$$|x|_{\beta} = \sqrt{x^2 + \beta}, \quad x \in \mathbb{R},$$

$$|v|_{\beta} = \sqrt{|v|^2 + \beta}, \quad v \in \mathbb{R}^2,$$

kde β je malý kladný parametr. Pak TV funkcionál přejde na

$$\int_{\Omega} \sqrt{|\nabla u| + \beta} dx = \int_{\Omega} |\nabla u|_{\beta} dx.$$

Model (5.41) pracuje dobře pro samotné odstranění šumu, ale zároveň velmi špatně kvůli rozmazávání, protože parametr λ volí uživatel sám, individuálně pro rozdílné obrázky a jejich poškození. Proto nebudeme řešit přímo (5.41), ale budeme hledat ustálené řešení nestacionární rovnice. Použijeme-li časově závislý model, s rostoucím t máme přístup k obnovené verzi obrázku a efektem evoluce může být detekce a zachování hran a zároveň vyhlazování šumu

$$u_t = -\lambda j * (j * u - u_0) + \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right). \quad (5.43)$$

Jedná se o evoluční parabolickou rovnici. V této formulaci předpokládáme *a priori* odhad Lagrangeova multiplikátoru λ , který představuje míru rozmazání použitého pro následné odstranění šumu. Tento evoluční proces je ale velmi pomalý a také zde platí přísná CFL podmínka

$$\frac{\Delta t}{\Delta x^2} \leq c|\Delta u| \quad (5.44)$$

pro pevné $c > 0$. Upravíme tedy model 5.41 do následujícího tvaru:

$$u_t = -|\nabla u| \lambda j * (j * u - u_0) + |\nabla u| \nabla \left(\frac{\nabla u}{|\nabla u|} \right). \quad (5.45)$$

Počátečními daty je poškozený (zašuměný, rozmazaný) obrázek u_0 , volíme homogenní Neumannovy okrajové podmínky.

Odvodíme nyní numerický model pro tuto rovnici v jedné dimenzi. Pak ∇u představuje partiální derivaci u_x . Eulerovu-Lagrangeovu rovnici v 1D

$$0 = - \left(\frac{u_x}{|u_x|} \right)_x + \lambda j * (j * u - u_0) \quad (5.46)$$

přepíšeme s parametrem $\beta > 0$:

$$0 = - \left(\frac{u_x}{|u_x|^\beta} \right)_x + \lambda j * (j * u - u_0). \quad (5.47)$$

Model v jedné dimenzi bude

$$u_t = -|u_x| \lambda j * (j * u - u_0) + \frac{\beta}{\beta + u_x^2} u_{xx}. \quad (5.48)$$

Parametr β hraje důležitější roli ve 2D modelu.

Stanovme hodnotu $\lambda > 0$ blízkou maximální hodnotě, kdy je explicitní schéma stabilní splňující CFL podmínku. Pro model ryziho odstranění šumu zjednodušíme konvoluci (popsáno vztahem (4.4)) ($j * (j * u - u_0)$) na $(u - u_0)$, tedy

$$u_t = -|\nabla u_x| \lambda (u - u_0) + \frac{\beta}{\beta + u_x^2} u_{xx}. \quad (5.49)$$

Nechť U_j^n je aproximace hodnoty $u(x_j, t_n)$, kde $x_j = j\Delta x$ a $t_n = n\Delta t$. Pro tuto rovnici získáme následující schéma

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = -|G_j| \lambda (U_j^n - u_0(x_j)) + \frac{\beta}{\beta + g_j^2} \frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{\Delta x^2}, \quad (5.50)$$

kde podle metody upwind

$$G_j = \frac{U_j^n - U_{j-1}^n}{\Delta x}$$

v případě, že $g_j(U_j^n - u_0(x_j)) > 0$ a

$$G_j = \frac{U_{j+1}^n - U_j^n}{\Delta x}$$

v případě, že $g_j(U_j^n - u_0(x_j)) < 0$, dále

$$g_j = \frac{U_{j+1}^n - U_{j-1}^n}{2\Delta x}.$$

Pro časovou derivaci použijeme explicitní Eulerovu metodu.

Vysvětleme si použití metody upwind pro (5.49) na jednodušší rovnici

$$q_t + f(q_x) = 0. \quad (5.51)$$

Zderivujeme rovnici podle x a získáme

$$(q_x)_t + [f(q_x)]_x = 0. \quad (5.52)$$

Zavedeme $u = q_x$, pak

$$u_t + [f(u)]_x = 0. \quad (5.53)$$

Pro $f(u) = a|u|$, kde a je konstanta, a $f'(u) = a \operatorname{sign}(u)$ máme rovnici (5.51) ve tvaru

$$q_t + a|q_x| = 0, \quad (5.54)$$

po prostorové derivaci a substituci $u = q_x$

$$u_t + a|u|_x = 0, \quad (5.55)$$

pak přepíšeme na

$$u_t + a \operatorname{sign}(u)u_x = 0. \quad (5.56)$$

To je důvod, proč využijeme metodu upwind, která volí dopřednou nebo zpětnou diferenci v závislosti na znaménku (podrobněji v sekci 3.3.1).

5.3.1 Rozšíření nelineární parabolické rovnice do 2D

Rozšíříme nyní rovnici (5.49) do dvou dimenzí

$$u_t = -\sqrt{u_x^2 + u_y^2} \lambda(u - u_0) + \frac{u_{xx}u_y^2 - 2u_{xy}u_xu_y + u_{yy}u_x^2}{u_x^2 + u_y^2}. \quad (5.57)$$

Při diskretizaci postupujeme obdobně jako u 1D rovnice. Necht $U_{i,j}^n$ je aproximace hodnoty $u(x_i, y_j, t_n)$, kde $x_i = i\Delta x$, $y_j = j\Delta y$ a $t_n = n\Delta t$. Pak můžeme numerické schéma napsat ve tvaru

$$\frac{U_{i,j}^{n+1} - U_{i,j}^n}{\Delta t} = -\sqrt{(G_{i,j}^x)^2 + (G_{i,j}^y)^2} \lambda(U_{i,j}^n - u_0(x_i, y_j)) + s_{i,j}^n, \quad (5.58)$$

kde

$$s_{i,j}^n = 0$$

pokud $(g_{i,j}^x)^2 + (g_{i,j}^y)^2 < \beta$ a

$$s_{i,j}^n = \frac{g_{i,j}^{xx}(g_{i,j}^y)^2 - 2g_{i,j}^{xy}g_{i,j}^xg_{i,j}^y + g_{i,j}^{yy}(g_{i,j}^x)^2}{(g_{i,j}^x)^2 + (g_{i,j}^y)^2}$$

jinak. Nyní použijeme centrální diference

$$\begin{aligned} g_{i,j}^x &= \frac{U_{i+1,j}^n - U_{i-1,j}^n}{2\Delta x}, \\ g_{i,j}^y &= \frac{U_{i,j+1}^n - U_{i,j-1}^n}{2\Delta y}, \\ g_{i,j}^{xx} &= \frac{U_{i+1,j}^n - 2U_{i,j}^n + U_{i-1,j}^n}{\Delta x^2}, \\ g_{i,j}^{yy} &= \frac{U_{i,j+1}^n - 2U_{i,j}^n + U_{i,j-1}^n}{\Delta y^2}, \end{aligned}$$

$$g_{i,j}^{xy} = \frac{U_{i+1,j+1}^n - U_{i-1,j+1}^n - U_{i+1,j-1}^n + U_{i-1,j-1}^n}{2\Delta x \Delta y}.$$

Další členy aproximujeme metodou upwind

$$G_{i,j}^x = \frac{U_{i,j}^n - U_{i-1,j}^n}{\Delta x}$$

v případě, že $g_{i,j}^x(U_{i,j}^n - u_0(x_i, y_j)) > 0$,

$$G_{i,j}^x = \frac{U_{i+1,j}^n - U_{i,j}^n}{\Delta x}$$

v případě, že $g_{i,j}^x(U_{i,j}^n - u_0(x_i, y_j)) < 0$,

$$G_{i,j}^y = \frac{U_{i,j}^n - U_{i,j-1}^n}{\Delta y}$$

v případě, že $g_{i,j}^y(U_{i,j}^n - u_0(x_i, y_j)) > 0$, a

$$G_{i,j}^y = \frac{U_{i,j+1}^n - U_{i,j}^n}{\Delta y}$$

v případě, že $g_{i,j}^y(U_{i,j}^n - u_0(x_i, y_j)) < 0$. Opět využijeme Eulerovu explicitní metodu pro časovou derivaci.

Vraťme se nyní k rovnici (5.57). Nahradíme závorku obsahující

$$(u - u_0)$$

dvojnásobnou konvolucí (4.4)

$$j * (j * u - u_0)$$

a tím přejdeme zpět k původnímu obecnějšímu problému. V numerickém schématu (5.58) nahradíme

$$(U_{i,j}^n - u_0(x_i, y_j))$$

výrazem

$$(W_{i,j}^n - V_0(x_i, y_j)),$$

kde W i V_0 jsou výsledkem jednoduché numerické metody

$$V_{i,j}^{n+1} = V_{i,j}^n - \frac{\Delta t}{\Delta x^2}(V_{i+1,j}^n - 2V_{i,j}^n + V_{i-1,j}^n) - \frac{\Delta t}{\Delta y^2}(V_{i,j+1}^n - 2V_{i,j}^n + V_{i,j-1}^n) \quad (5.59)$$

po dvou, resp. jednom časovém kroku; počátečními hodnotami cyklů volíme $Q_{i,j}^n$, resp. $u_0(x_i, y_j)$.

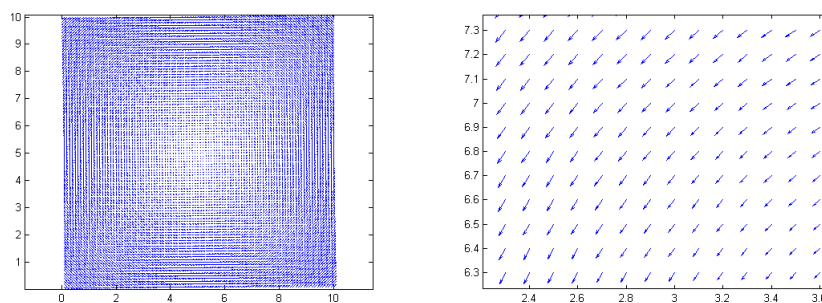
Kapitola 6

Numerické experimenty

V této kapitole aplikujeme dříve uvedené metody na digitální obrázky. Ve většině případů pracujeme pouze se šedotónovými obrázky, ale pro ukázkou se seznámíme i s aplikací na barevné obrázky. Výsledky získané modely založenými na parciálních diferenciálních rovnicích (kapitola 5) porovnáme také s výstupy programu, ve kterém byly použity klasické metody (kapitola 4). Seznam použitých kódů naleznete v příloze A.

6.1 Rozmazání spojené se změnou souřadnic/pozice

Vlnová (transportní) rovnice (3.3), což je nejčastěji jmenovaný zástupce hyperbolických parciálních diferenciálních rovnic, je obecně známá šířením informace rychlostí a . Ve dvou dimenzích (rovnice (3.6)) máme k dispozici dva koeficienty a a b , které mohou být konstantní, pak se informace šíří do směrů x a y podle vektoru $[a, b]$, nebo prostorově proměnné, například mohou popisovat rotační vektorové pole (znázorněno na obr. 6.1). Jelikož používáme metodu upwind (sekce 3.3.1), která je prvního řádu přesnosti v čase i prostoru, dochází k výraznému vyhlazování dat. To bychom mohli zlepšit například použitím metody vyššího řádu. Zobrazíme si původní data jako trojrozměrný obrázek, kde si třetí rozměr (výšku), můžeme představit jako hodnotu jasu.



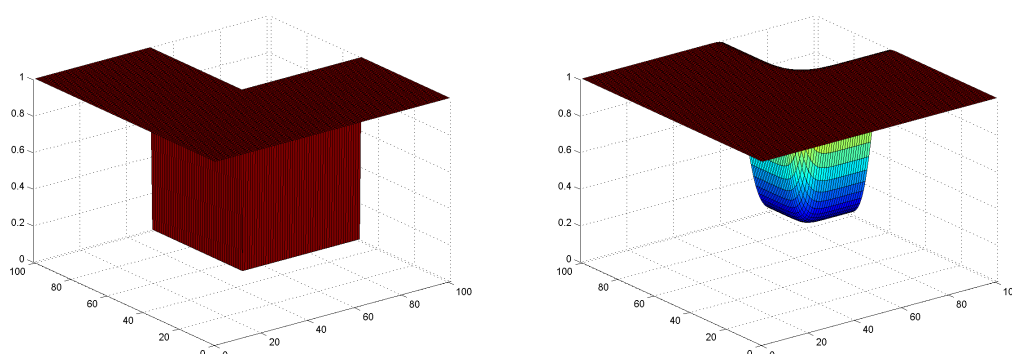
Obr. 6.1: Rotační vektorové pole, na pravém obrázku je přiblížena část 2. kvadrantu.

Nejprve volíme konstantní parametry a a b , tedy objekt se bude v čase přemísťovat konstantní rychlostí podle složek vektoru $[a, b]$ a také bude docházet k zaoblení hran. Původní stav a situace po sto časových krocích jsou znázorněny na obr. 6.2. Velikosti kroků Δx , Δy a Δt a parametry a a b jsou voleny tak, aby splňovaly CFL podmínku

$$|a| \frac{\Delta t}{\Delta x} + |b| \frac{\Delta t}{\Delta y} \leq 1, \quad (6.1)$$

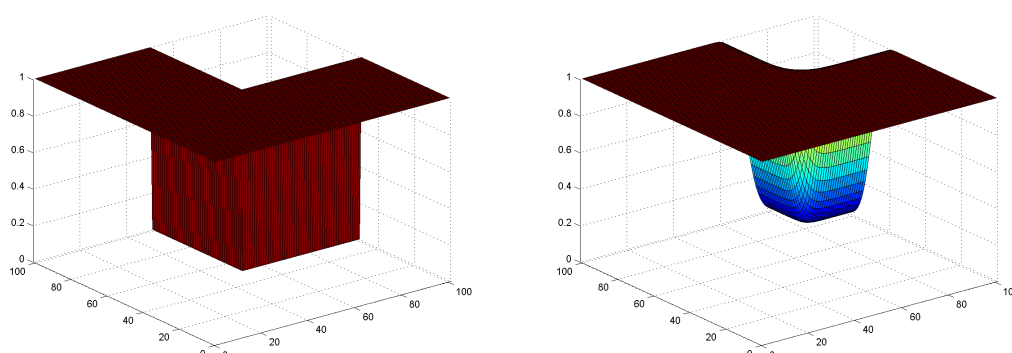
v tomto případě $a = b = 2$, $\Delta x = \Delta y = 0.1$, $\Delta t = 0.01$. Simulace je spustitelná z hlavního souboru *hyperbolic_posun_main.m*, který vytvoří jednoduchý objekt (využijeme 3D vykreslení, kde třetí rozměr odpovídá jasů - barvě) a vykreslí původní a vypočtená data. Samotný výpočet je připraven v souboru *hyperbolic_posun_advection.m*, využíváme zde předdefinované funkce *ode45()* programu MATLAB (podrobný popis předdefinovaných funkcí programu MATLAB naleznete např. v odkazu [18]); jedná se o explicitní metodu. Pokud bychom zvýšili počet časových kroků, řešení by se limitně přiblížilo k hodnotě 1 na celé oblasti.

Volíme homogenní Neumannovu podmínku, kterou realizujeme kopírováním hodnot z druhého/předposledního řádku a sloupce matice reprezentující obrázek do prvního/posledního řádku a sloupce. Stejně tomu bude i ve všech následujících odstavcích (důvod této volby vysvětlíme později).



Obr. 6.2: Hyperbolická rovnice - posunutí; vlevo počáteční stav, vpravo situace po 100 krocích.

Stejný je i výpočet pomocí numerických toků (obr. 6.3). Hlavním souborem obsahujícím tvorbu počátečního objektu a vykreslení je *hyperbolic_posun_toky_main.m*, který opět využívá funkce *ode45()* (tento soubor byl vytvořen pro snadnou modifikaci na jiné verze a schémata), numerické schéma je vytvořeno v souboru *hyperbolic_advection.m* a tokové funkce jsou popsány v souborech *hyperbolic_funF.m* a *hyperbolic_funG.m*. Parametry jsou voleny stejně jako v předchozím případě.



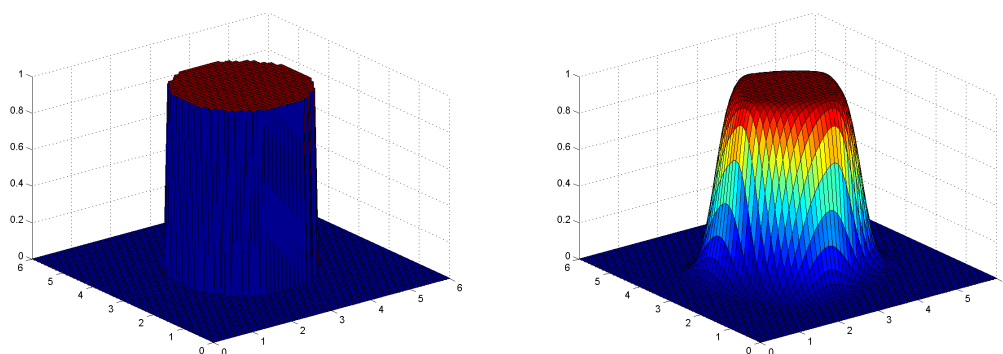
Obr. 6.3: Hyperbolická rovnice - posunutí; vypočteno pomocí numerických toků; vlevo počáteční stav, vpravo situace po 100 krocích.

Pro prostorově proměnné parametry ([19])

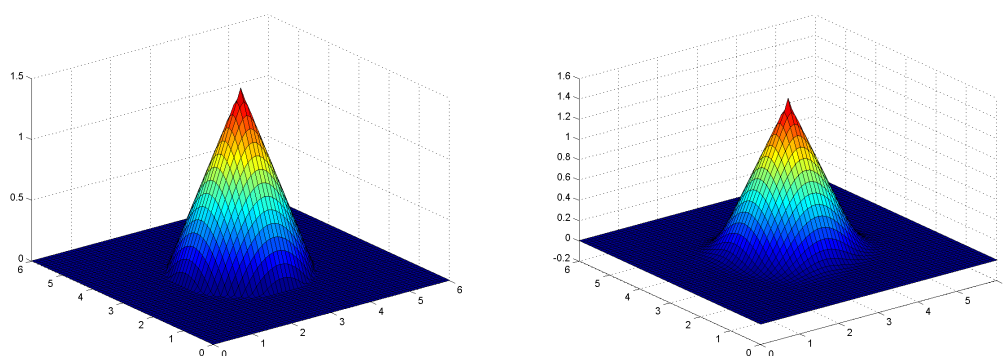
$$a = 2\pi\Delta x(y - y_s), \quad (6.2)$$

$$b = -2\pi\Delta y(x - x_s), \quad (6.3)$$

kde x a y jsou souřadnice konkrétního bodu a x_s a y_s jsou souřadnice středu obrázku, kolem kterého celé pole rotuje, budou původní data rotovat kolem středu obrázku konstantní úhlovou rychlostí (obr. 6.1). Opět bude docházet k vyhlazování. Původní a rozmazaný obrázek pro různé počáteční podmínky (kužel, válec a šikmo seříznutý válec, na kterém je dobře patrné pootočení) jsou vykresleny níže (obr. 6.4, 6.5, 6.6). Soubor, ve kterém je uložena hlavní část kódu, je pojmenován *hyperbolic_main.m*, dále k výpočtu potřebujeme *hyperbolic_advection.m*, *hyperbolic_funF.m* a *hyperbolic_funG.m*, jejichž úloha je popsána výše. Opět využíváme funkce *ode45()*. Parametry jsou samozřejmě voleny tak, aby splňovaly CFL podmínku, tedy $\Delta x = \Delta y = 0.1$, $\Delta t = 0.001$ a parametry a a b dosahují maximálně hodnoty $\frac{6}{10}\pi$. Obrázky vpravo ukazují stav po tisíci krocích.

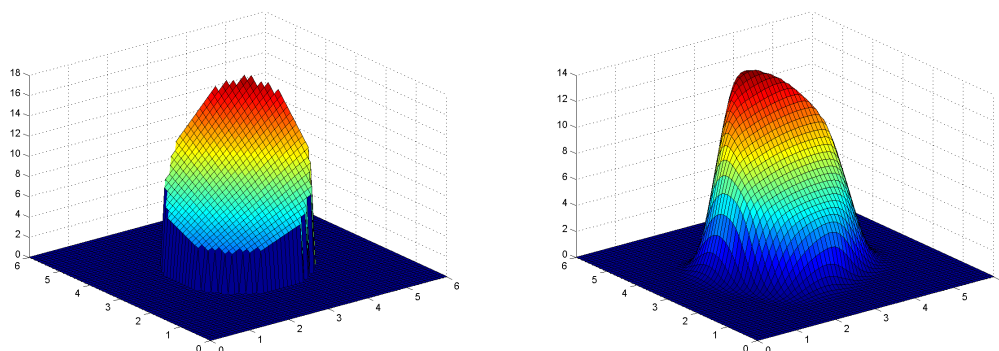


Obr. 6.4: Hyperbolická rovnice - rotace válce; vlevo počáteční stav, vpravo pootočené a rozmazané těleso.



Obr. 6.5: Hyperbolická rovnice - rotace kužele; vlevo počáteční stav, vpravo pootočené a rozmazané těleso.

Čtenáře jistě zajímá, jaký výsledek získáme pro dvourozměrný obrázek, nikoliv pro trojrozměrný graf. Na obrázku 6.7 je ukázka šedotónového obrázku, vyhlazování hran patrné ve 3D zobrazení se projeví rozmazáním obrázku. Výpočet je uložen v hlavním souboru *image_gray_main.m* (obsahuje nastavení parametrů, načtení šedotónového obrázku a vykreslení, pomocí tohoto souboru spustíme výpočet); soubor *image_advection.m* slouží k samotnému výpočtu, využíváme zde předdefinované funkce *ode45()*, tedy jedná se o explicitní metodu.

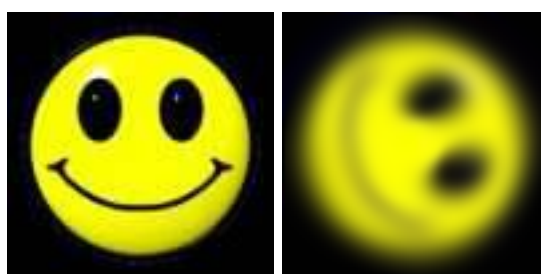


Obr. 6.6: Hyperbolická rovnice - rotace středově nesouměrného tělesa (seříznutého válce); vlevo počáteční stav, vpravo pootočené a rozmazané těleso.



Obr. 6.7: Hyperbolická rovnice - šedotónový obrázek; vlevo původní stav, vpravo rozmazaný obrázek.

Stejný postup vyzkoušíme i na barevný obrázek se třemi barevnými složkami RGB. Postupujeme stejným způsobem pro každou barevnou složku zvlášť, poté barevné složky znovu složíme dohromady a vykreslíme (obr. 6.8). Pro výpočet potřebujeme obrázek a soubory *image_rgb_main.m* a *image_advection.m*.



Obr. 6.8: Hyperbolická rovnice - barevný obrázek; vlevo původní stav, vpravo rozmazaný obrázek.

Dále budeme zkoumat jen šedotónové obrázky, případně trojrozměrný model.

6.2 Rozmazání bez pohybu obrázku

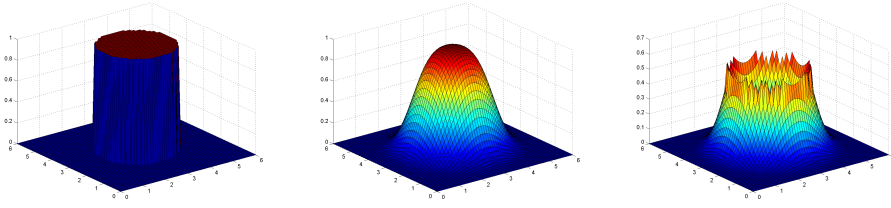
Předchozí odstavce se týkaly hyperbolické rovnice, která v závislosti na koeficientech a a b šířila informace daným směrem. Současně ale docházelo k posunutí nebo rotaci počátečních dat, což je v mnoha případech zpracování obrazu nežádoucím jevem. Zaměříme nyní pozornost na lineární parabolickou rovnici ve dvou dimenzích (5.26)

$$q_t + (aq_x)_x + (bq_y)_y = 0, \quad (6.4)$$

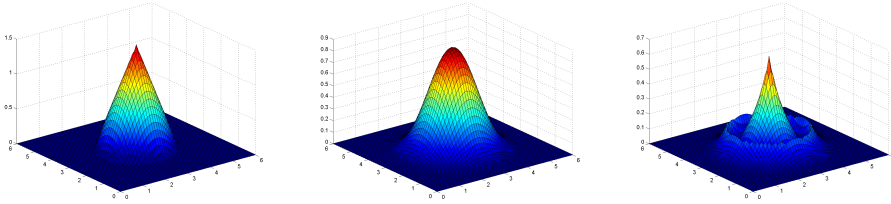
parametry a a b volíme konstantní záporné, pak rovnice přejde v (5.27)

$$q_t + aq_{xx} + bq_{yy} = 0. \quad (6.5)$$

Nejprve se blíže podíváme na explicitní metodu (5.29). Aby byla metoda stabilní, musí parametry splňovat CFL podmínku (5.34), v souboru *parabolic_expl_main.m* volíme $\Delta x = \Delta y = \frac{1}{10}$, $\Delta t = \frac{1}{1000}$, $a = b = -\frac{25}{10}$. Výpočet je realizován funkcí *ode45()* v pomocném souboru *parabolic_advection.m*, tokové funkce jsou rozepsané v souborech *parabolic_funF.m* a *parabolic_funG.m*. Na obrázcích 6.9, 6.10 a 6.11 jsou znázorněna jednoduchá tělesa (jedná se opět o trojrozměrný náhled, třetí rozměr reprezentuje hodnotu jasu) a chyba metody určená rozdílem vypočtených a počátečních hodnot v absolutní hodnotě. Obrázky vlevo znázorňují počáteční stav, uprostřed je zobrazena situace po padesáti časových krocích a vpravo je zobrazena chyba metody.



Obr. 6.9: Parabolická rovnice - válec; CFL 1/2; vlevo počáteční stav, uprostřed stav po 50 krocích, vpravo chyba metody



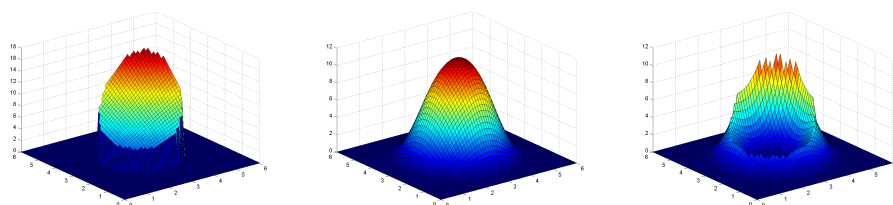
Obr. 6.10: Parabolická rovnice - kužel; CFL 1/2; vlevo počáteční stav, uprostřed stav po 50 krocích, vpravo chyba metody.

Lepších výsledků dosáhneme, když zpřísníme CFL podmínku (5.34) (pravou stranu nerovnosti omezíme hodnotou $\frac{1}{6}$). V tomto případě volíme stejné parametry Δx , Δy a Δt jako v předchozím případě, změníme jen váhu koeficientů a a b na hodnotu $a = b = -\frac{10}{6}$ (obrázky 6.12, 6.13 a 6.14).

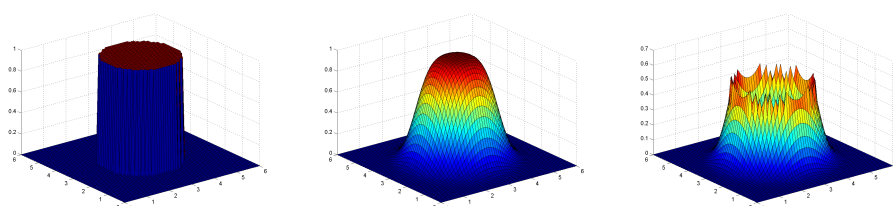
Porovnáním výsledků pro CFL podmínku $\frac{1}{2}$ a $\frac{1}{6}$ vidíme, že rozmazávání je v druhém případě menší (a tedy i chyba) - patrné například na obrázcích s válcem (6.9, 6.12) nebo se středově nesouměrným tělesem (6.11, 6.14).

Pro zajímavost si ukažme, jak bude numerické schéma reagovat na prostorově proměnné parametry a a b , zvolíme-li je podobně jako v sekci 6.1, tedy

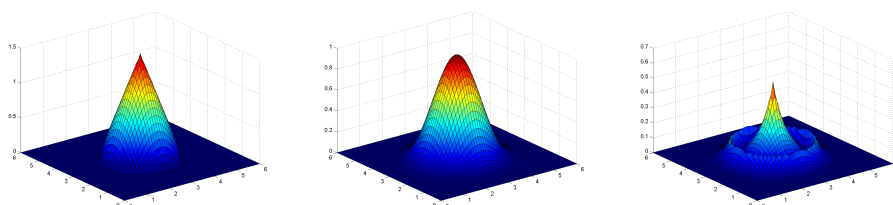
$$a = -2\pi\Delta x(y - y_s)\frac{7}{9}, \quad (6.6)$$



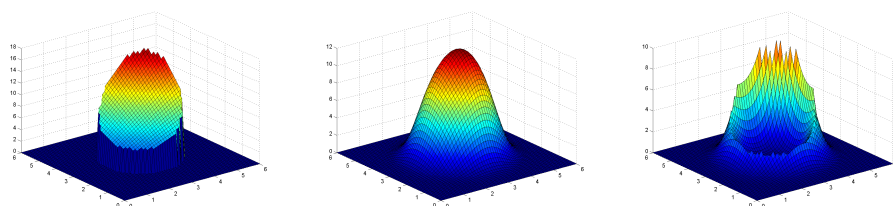
Obr. 6.11: Parabolická rovnice - středově nesouměrné těleso (seříznutý válec); CFL 1/2; vlevo počáteční stav, uprostřed stav po 50 krocích, vpravo chyba metody.



Obr. 6.12: Parabolická rovnice - válec; CFL 1/6; vlevo počáteční stav, uprostřed stav po 50 krocích, vpravo chyba metody.



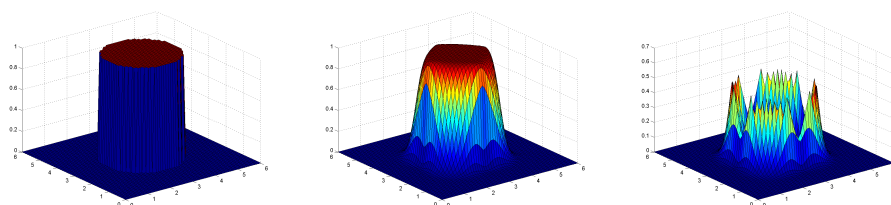
Obr. 6.13: Parabolická rovnice - kužel; CFL 1/6; vlevo počáteční stav, uprostřed stav po 50 krocích, vpravo chyba metody.



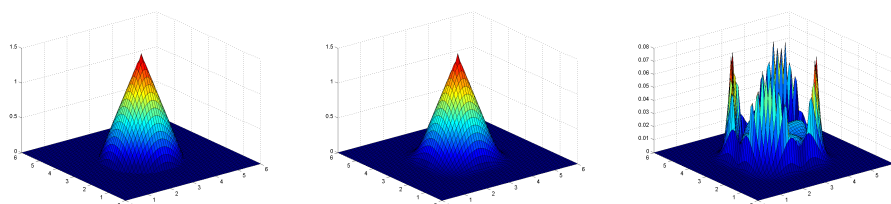
Obr. 6.14: Parabolická rovnice - středově nesouměrné těleso (seříznutý válec); CFL 1/6; vlevo počáteční stav, uprostřed stav po 50 krocích, vpravo chyba metody.

$$b = -2\pi\Delta y(x - x_s)\frac{7}{9}, \quad (6.7)$$

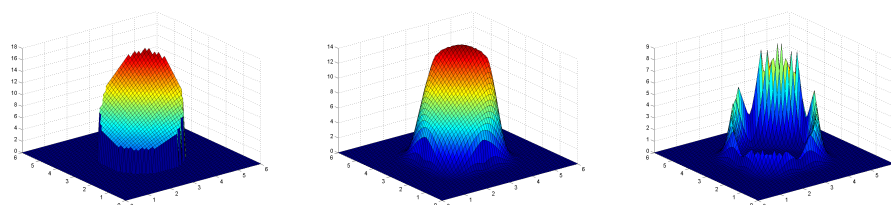
konstanta $\frac{7}{9}$ nám pouze kontroluje, abychom neporušili CFL podmínku (5.34). Na obrázcích 6.15, 6.16 a 6.17 si všimněme výraznějších změn ve směrech os x a y .



Obr. 6.15: Parabolická rovnice - válec; nekonzstantní parametry a, b ; vlevo počáteční stav, uprostřed stav po 50 krocích, vpravo chyba metody.



Obr. 6.16: Parabolická rovnice - kužel; nekonzstantní parametry a, b ; vlevo počáteční stav, uprostřed stav po 50 krocích, vpravo chyba metody.



Obr. 6.17: Parabolická rovnice - středově nesouměrné těleso (seříznutý válec); nekonzstantní parametry a, b ; vlevo počáteční stav, uprostřed stav po 50 krocích, vpravo chyba metody.

V sekci 5.2 jsme se zmínili o možnosti využití implicitního numerického schématu pro parabolickou rovnici ve dvou prostorových dimenzích. V souboru *parabolic_impl_main.m* zaměníme explicitní funkci *ode45()* za implicitní variantu *ode15s()* ([20]). Pomocné soubory zůstávají neměnné. Nezpochybnitelnou výhodou implicitních metod je stabilita pro libovolnou volbu a vzájemnou vazbu parametrů Δt , Δx , Δy , a a b , nevýhodou je složitější programování a také vysoké nároky na paměť počítače. Pro stejnou volbu parametrů jako při použití explicitní metody dospějeme ke stejným výsledkům.

6.3 Odstranění šumu se zachováním hran v 1D

V předchozích sekcích jsme se seznámily s technikami, které dokáží obrázek rozmazat - to může být jednou z cest, jak se zbavit nežádoucího šumu. Zároveň s odstraněním šumu ale dochází i k rozmazávání hran a mizí detaily. Zkusme nyní využít získaných znalostí a spojit je s myšlenkou detektorů hran (sekce 4.1.4), kdy v obraze hledáme výraznou nespojitost jasu; v takovém místě očekáváme hranu. Shrňme-li celou myšlenku do jedné věty: pro pixely v místech nespojitosti jasu zachováváme původní jasové hodnoty, jinde obrázek mírně rozmážeme.

Nejprve si aplikaci metody (numerické schéma (5.50)) uvedené v sekci 5.3 blíže prohlédneme

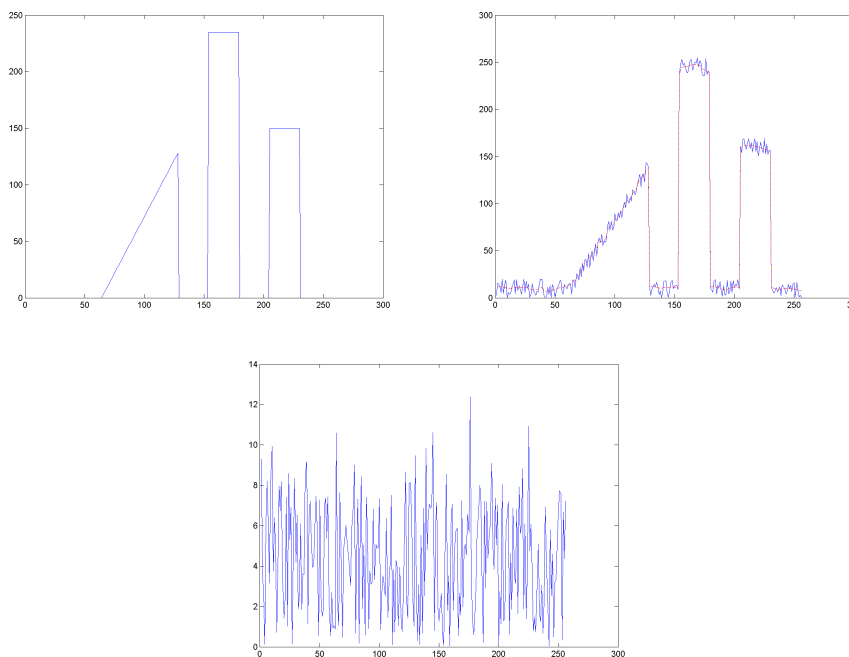
v jedné dimenzi. Původní obrázek obsahuje spojitou po částech lineární funkci (v 1D označujeme výrazem „signál“; obr. 6.18 vlevo) s hodnotami v intervalu $(0, 235)$, ke kterým přičítáme náhodné číslo z intervalu $(0, 20)$ (matlabovská funkce `rand()`), které představuje šum. Na obrázku 6.18 vpravo vidíme modře vykreslený signál se šumem a červeně vykreslené řešení (soubor `parabolic_nelin_1D.m`, parametry $\Delta x = 1$, $\Delta t = \frac{1}{4}$, 50 časových kroků, $\lambda = 0.25$, $\beta = 10$). Parametry λ a β jsou voleny takto kvůli splnění CFL podmíněk

$$\frac{\Delta t}{\Delta x^2} \leq \frac{\beta + u_x^2}{2\beta} \quad (6.8)$$

pro parabolický člen a

$$\frac{\Delta t}{\Delta x} \leq c\lambda \sqrt{1 + \frac{\beta}{u_x^2}} \quad (6.9)$$

pro konvektivní člen (pro pevné c), protože pro časovou derivaci používáme Eulerovo explicitní schéma. Na obrázku 6.18 dole je vykreslen absolutní rozdíl hodnot jasu obrázku se šumem a řešení popsané numerické metody pro uvedené parametry. Všimněme si, že metoda nevytváří nežádoucí *stair-case effect* (sekce 5.3).



Obr. 6.18: Odstranění šumu v 1D; vlevo původní signál, vpravo signál se šumem (modře) a výsledek metody (červeně), dole je znázorněna chyba metody.

6.4 Odstranění šumu se zachováním hran v 2D

V předchozí sekci jsme vyzkoušeli, jak metoda popsaná v sekci 5.3 dokáže vyhladit poškozený signál aniž by rozmazala hrany/nespojivosti. Upravenou metodu (5.58) pro dvourozměrná data aplikujeme na zašuměný obrázek.

Existuje celá řada šumů, které dokážeme matematicky popsat. Seznamme se s vybranými šумы, které se v této práci pokusíme odstranit nebo alespoň redukovat:

- *Gaussovský šum* - je popsán střední hodnotou a rozptylem; každý pixel obrazu je mírně pozměněn (obr. 6.19 vpravo nahoře)
- *Gaussovský bílý šum* - náhodný signál s normálním rozložením ([21]), popsán hodnotou SNR (*signal-to-noise ratio*) (obr. 6.19 vlevo dole)
- *Sůl a pepř* (angl. *salt and pepper*) - šum tvoří náhodně rozmístěné černé a bílé pixely (obr. 6.19 vpravo dole)

Jednotlivé varianty šumu byly uměle přidány do původního obrázku známém po celém světě pod názvem Lena (soubor *lena_original.bmp*) v programu MATLAB.



Obr. 6.19: Ukázky šumu - původní obrázek (vlevo nahoře, převzato z [22]); Gaussovský šum (vpravo nahoře, střední hodnota je rovna nule, rozptyl je roven $\frac{1}{100}$); Gaussovský bílý šum (vlevo dole, $SNR = 3$); šum sůl a pepř (vpravo dole, hustota $\frac{5}{100}$).

Pro numerické experimenty v této sekci volíme následující parametry: $\Delta x = \Delta y = 1$, $\Delta t = \frac{1}{10}$, 50 časových kroků, $\beta = 10^{-7}$. Nevýhodou této metody je skutečnost, že parametr λ se liší v závislosti na zašumění a zvoleném obrázku. Doporučuje se volit λ takové, aby byly splněny CFL podmínky (6.8) a (6.9) rozšířené do dvou dimenzí. Soubor *parabolic_nonlin_2D.m* pro tento výpočet nepotřebuje žádné pomocné soubory, je připravena řada testovacích obrázků (jejich seznam naleznete v příloze A).

Ukažme si, jak citlivá je volba parametru λ . Srovnajme původní obrázek 6.19 vlevo nahoře s řešením pro Gaussovský bílý šum ($SNR = 3$) (obr. 6.20 vlevo nahoře) a parametr

- $\lambda = 1$ (obr. 6.20 vpravo nahoře) - dochází k přílišnému rozmazání a ztrácí se detaily,
- $\lambda = 8$ (obr. 6.20 vlevo dole) - šum není dostatečně vyhlazen,

- $\lambda = 9$ (obr. 6.20 vpravo dole) - kvůli nesplnění CFL podmínky není metoda stabilní.



Obr. 6.20: Srovnání výstupů programu pro Gaussovský bílý šum ($SNR = 3$) - Lena - vstupní obrázek se šumem (nahore vlevo); řešení pro $\lambda = 1$ (vpravo nahore); řešení pro $\lambda = 8$ (vlevo dole); řešení pro $\lambda = 9$ (vpravo dole).

Nyní porovnáme výsledky pro Gaussovský šum se střední hodnotou rovnou nule a rozptylem $\frac{1}{100}$ (obr. 6.19 vpravo nahore), parametr

- $\lambda = 2$ (obr. 6.21 vpravo nahore) - obrázek je rozmazaný a ztrácí se detaily,
- $\lambda = 12$ (obr. 6.21 vlevo dole) - šum je odstraněný a nedochází k výraznému rozmazání,
- $\lambda = 13$ (obr. 6.21 vpravo dole) - kvůli nesplnění CFL podmínky není metoda stabilní.

Pro šum sůl a pepř s hustotou $\frac{5}{100}$ (obr. 6.19 vpravo dole) volíme parametr λ následovně:

- $\lambda = 1$ (obr. 6.22 vpravo nahore) - šum je odstraněný, nedochází k rozmazání,
- $\lambda = 7$ (obr. 6.22 vlevo dole) - šum není dokonale vyhlazen,
- $\lambda = 8$ (obr. 6.22 vpravo dole) - kvůli nesplnění CFL podmínky není metoda stabilní (v místě peří na klobouku).



Obr. 6.21: Srovnání výstupů programu pro Gaussovský šum - Lena - vstupní obrázek se šumem (nahore vlevo); řešení pro $\lambda = 2$ (vpravo nahore); řešení pro $\lambda = 12$ (vlevo dole); řešení pro $\lambda = 13$ (vpravo dole).

Odvozenou metodu použijeme i na jiné obrázky - jako první to bude šedotónová varianta slavné malby Mona Lisa od Leonarda da Vinciho (převzato z [11]), dále kameraman (převzato z [23]), který je stejně jako Lena často testovaný počítačovými grafiky. Parametry Δt , Δx , Δy , β a počet časových kroků zůstává stejný, mění se pouze parametr λ a to následovně:

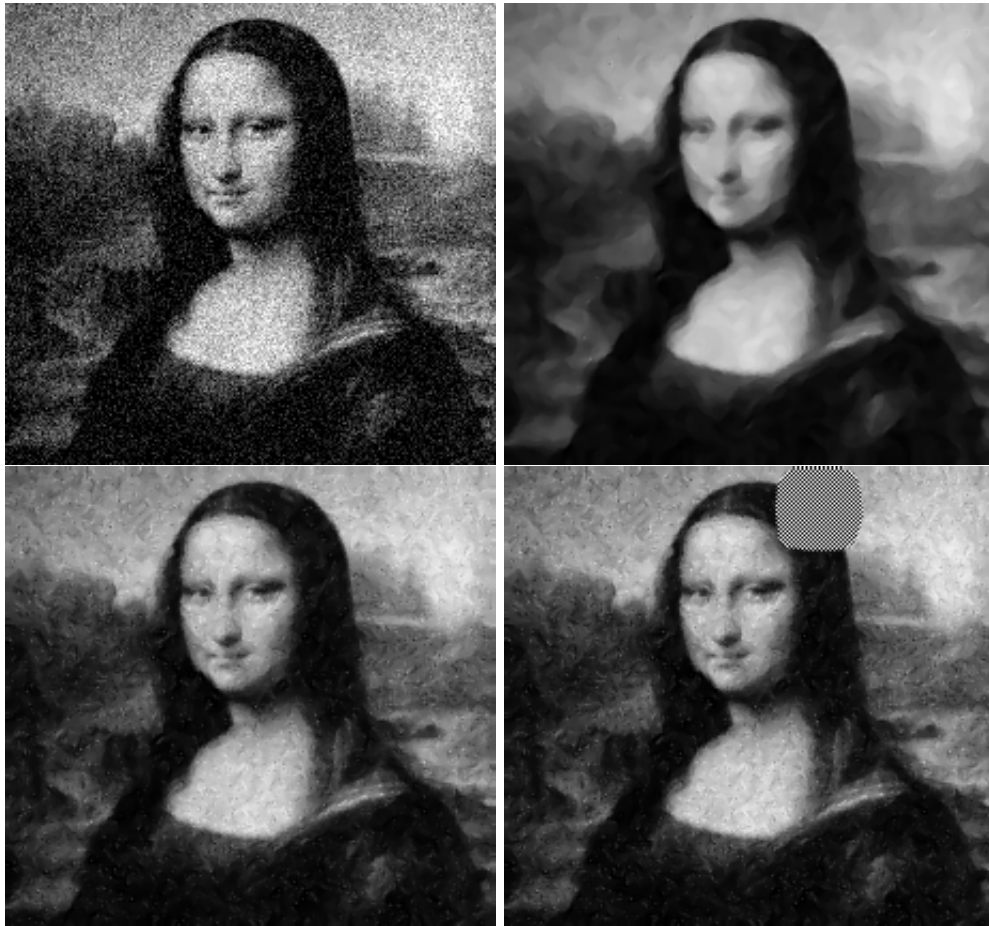
- Mona Lisa
 - $\lambda = 4$ (obr. 6.23 vpravo nahore) - velmi rozmazané,
 - $\lambda = 14$ (obr. 6.23 vlevo dole) - není výrazně rozmazané, ale šum není úplně vyhlazen
 - $\lambda = 15$ (obr. 6.23 vpravo dole) - pro tuto hodnotu parametru λ je metoda nestabilní,
- kameraman
 - $\lambda = 2$ (obr. 6.24 vpravo nahore) - velmi rozmazané,
 - $\lambda = 8$ (obr. 6.24 vlevo dole) - méně rozmazané, šum je téměř odstraněný,
 - $\lambda = 12$ (obr. 6.24 vpravo dole) - pro tuto hodnotu parametru λ je metoda nestabilní.



Obr. 6.22: Srovnání výstupů programu pro šum sůl a pepř (s hustotou $\frac{5}{100}$) - Lena - vstupní obrázek se šumem (nahore vlevo); řešení pro $\lambda = 1$ (vpravo nahore); řešení pro $\lambda = 7$ (vlevo dole); řešení pro $\lambda = 8$ (vpravo dole).

V obou případech je obrázek poškozen Gaussovským šumem s nulovou střední hodnotou a rozptylem $\frac{1}{100}$ (obr. 6.23 a 6.24 vlevo nahore).

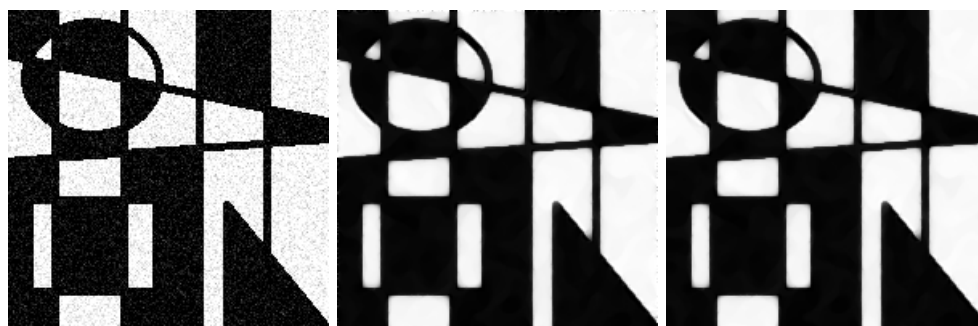
Vrátme se nyní k volbě okrajové podmínky zmíněné v sekci 6.1. Na obrázku 6.25 uprostřed je znázorněn výběr nevhodné okrajové podmínky (konkrétně zde volíme okrajové podmínky stejné jako počáteční, proto se v prvním/posledním řádku a sloupci zachovává šum). To je důvod, proč jsme zvolili homogenní Neumannovu okrajovou podmínku realizovanou kopírováním hodnot z druhého/předposledního řádku a sloupce do prvního/posledního (obr. 6.25 vpravo). Na obrázku 6.26 je přiblížen pravý horní roh obrázku 6.25 uprostřed s dobře viditelnými nevyhlazenými pixely. Tento problém nastává hlavně pro černobílé obrázky (například šachovnice), u jiných obrázků to není tak patrné.



Obr. 6.23: Srovnání výstupů programu pro Gaussovský šum - Mona Lisa - vstupní obrázek se šumem (nahore vlevo); řešení pro $\lambda = 4$ (vpravo nahore); řešení pro $\lambda = 14$ (vlevo dole); řešení pro $\lambda = 15$ (vpravo dole).



Obr. 6.24: Srovnání výstupů programu pro Gaussovský šum - kameraman - vstupní obrázek se šumem (nahore vlevo); řešení pro $\lambda = 2$ (vpravo nahore); řešení pro $\lambda = 8$ (vlevo dole); řešení pro $\lambda = 12$ (vpravo dole).



Obr. 6.25: Vysvětlení volby okrajové podmínky - obrázek se šumem (vlevo); obrázek po odstranění šumu s nevhodně zvolenými okrajovými podmínkami (uprostřed); obrázek po odstranění šumu s homogenními Neumannovými okrajovými podmínkami (vpravo).



Obr. 6.26: Vysvětlení volby okrajové podmínky - přiblížený pravý horní roh, kde jsou dobře viditelné nevyhlazené počáteční podmínky.

6.4.1 Odstranění šumu a detekce hran - jiné přístupy

V kapitole 4 jsme se seznámili s několika možnostmi, jak vyhlazovat šum pomocí filtračních masek. Příklady filtrů naleznete v sekci 4.1.4. Obrázek 6.27 ukazuje výstupy programu PYTHON pro data poškozená Gaussovským šumem (obr. 6.27 vpravo nahoře) - rovnoměrná maska (vlevo uprostřed), Gaussova maska (vpravo uprostřed), medián (vlevo dole), bilaterální filtr (vpravo dole). Software používaný pro zpracování obrazu (jmenujme například MATLAB, PYTHON) většinou zahrnuje několik předdefinovaných filtrů, další je možné snadno vytvořit - popsat maticemi a 2D konvolucí stanovit řešení.

Na obrázku 6.28 jsou vykresleny výstupy programu PYTHON pro šum označovaný výrazem sůl a pepř, využíváme stejné masky jako v předchozím případě.

Další možností, jak redukovat šum, je Fourierova metoda, kdy odstraníme vysoké frekvence (tzv. dolní propust) - popsáno v sekci 4.1.3. Původní a zašuměná data jsou vykreslena na obr. 6.29 vlevo a vpravo nahoře (opět se jedná o Gaussovský šum), vlevo dole vidíme výstup programu po odstranění vysokých frekvencí a vpravo dole je zobrazen odstraněný šum.

Jedná se o přímé použití jmenovaných masek po celé ploše obrázku. Stejně jako v případě zkoumaných numerických schémat by bylo vhodné aplikovat masky pouze v místech, kde nejsou hrany. Pro obrázek *lena_original.bmp* jsme detekovali hrany pomocí vybraných gradientních operátorů (Canny, Laplace, Sobel - podrobněji v sekci 4.1.4), výsledky jednotlivých detektorů hran můžeme porovnat na obr. 6.30. Detekce hran byla realizována v softwaru PYTHON předdefinovanými funkcemi, podobné možnosti nabízí i MATLAB s funkcí *edge()* s volitelnými parametry 'sobel', 'prewitt', 'roberts', 'canny' Další gradientní operátory lze vytvořit konvolucí obrázku a matic představujících gradientní operátory v daných směrech. Pro masku Sobel definujeme zvlášť hrany ve směru os x a y . Pro zlepšení výsledků by bylo třeba naprogramovat metodu, která by v blízkosti hran zabránila aplikaci vyhlazovacích masek.



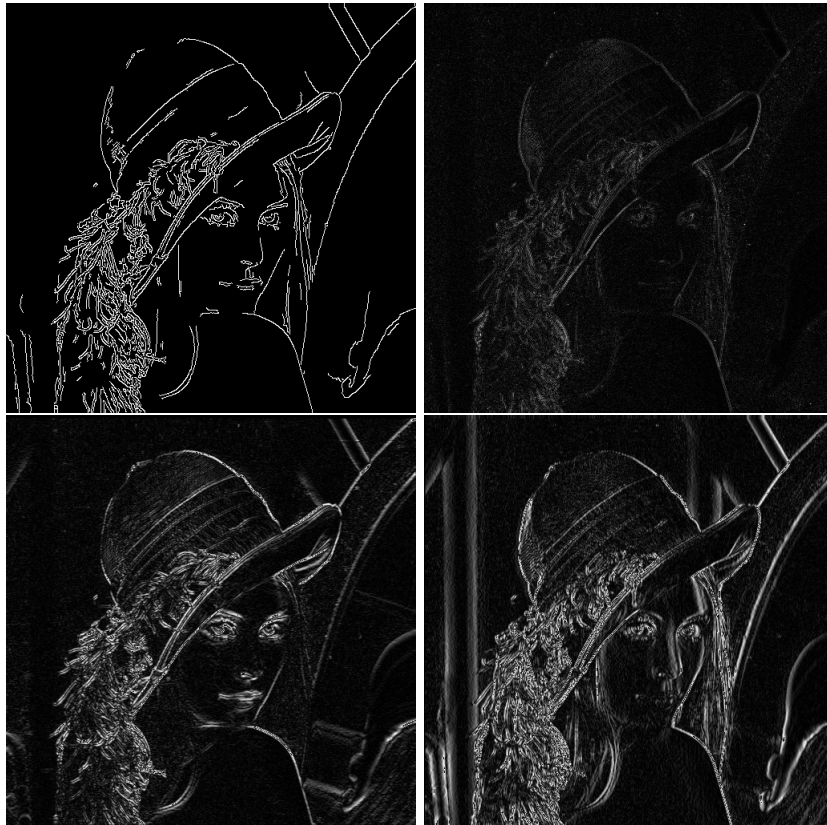
Obr. 6.27: Ukázky průměrování pro obrázek poškozený Gaussovským šumem - originální obrázek (vlevo nahoře); obrázek poškozený Gaussovským šumem (vpravo nahoře); rovnoměrná maska (vlevo uprostřed); Gaussova maska (vpravo uprostřed, [24]); medián (vlevo dole); bilaterální filtr (vpravo dole, [24]).



Obr. 6.28: Ukázky průměrování pro obrázek poškozený šumem „sůl a pepř“ - originální obrázek (vlevo nahoře); obrázek poškozený Gaussovským šumem (vpravo nahoře); rovnoměrná maska (vlevo uprostřed); Gaussova maska (vpravo uprostřed, [24]); medián (vlevo dole); bilaterální filtr (vpravo dole, [24]).



Obr. 6.29: Odstranění šumu - frekvenční analýza - originální obrázek (vlevo nahoře); obrázek poškozený Gaussovským šumem (vpravo nahoře); obrázek po odstranění vysokých frekvencí (vlevo dole); vysoké frekvence - šum (vpravo dple).



Obr. 6.30: Ukázky detekce hran - Canny (vlevo nahoře); Laplace (vpravo nahoře); Sobel vyhledávající hrany ve směru x (vlevo dole); Sobel vyhledávající hrany ve směru y (vpravo dole).

6.4.2 Pozorování

Podíváme-li se pozorně na obrázky v předchozích sekcích, všimneme si, že numerická metoda testovaná v sekci 6.4 dává rozdílné výsledky v závislosti na parametru λ . Nejlepší volbou se jeví co největší hodnota tohoto parametru, pro který je metoda stabilní, ale i přesto dochází u některých obrázků k přílišnému rozmazání. Tomu lze částečně zabránit například zmenšením počtu časových kroků.

Už samotná skutečnost, že volba parametru není přesně dána a závisí na mnoha faktorech (struktúře obrázku, přidanému šumu...), je velkou nevýhodou metody. Také zvolené explicitní Eulerovo schéma nás omezuje při volbě λ - abychom nebyli svazováni CFL podmínkou, museli bychom naprogramovat implicitní numerické schéma, což by ale bylo vzhledem k nelineárním členům rovnice komplikované a také by se výpočet zpomalil. Přesto se nám podařilo nalézt počáteční data a parametry, pro které metoda dává uspokojivé výsledky. Ukázalo se, že volbou λ můžeme ovlivňovat míru rozmazání podle dalšího využití výsledků - někdy pro další zpracování potřebujeme zachovat ostré hrany, jindy preferujeme úplné vyhlazení šumu a nevádí nám ztráta rozmazaných detailů.

Vyhlazovací masky nemůžeme s touto metodou přímo srovnávat, protože při výpočtu nebereme ohled na existující hrany. Při vhodné kombinaci s hranovými detektory a přiměřeném vícenásobném opakování konvoluce masek s obrázkem bychom dosáhli lepších výsledků.

Kromě Gaussovského a Gaussovského bílého šumu jsme vstupní obrázek poškodili i šumem zvaným sůl a pepř, pro který se ukázalo, že nejjednodušším a nejefektivnějším přístupem je volba mediánu (obr. 6.28 vlevo dole), protože numerická metoda tento šum úplně nevyhladí a navíc numerická metoda je časově i implementačně mnohem náročnější.

Kapitola 7

Závěr

Cílem práce bylo porovnání přístupů pro zpracování digitalizovaného obrazu. Při realizaci této práce jsme vyzkoušeli aplikaci několika metod na digitální obrazy, které byly poškozeny různými druhy šumu, a sledovali jsme, jak tyto metody řeší daný problém.

Nejprve jsme ukázali, jaký vliv mají numerická schémata lineárních parciálních diferenciálních rovnic na rozmazání jednoduchých geometrických tvarů a obrázků, protože rozmazání je základním způsobem redukce šumu. Jejich rozšířením o nelineární člen způsobilo, že kromě vyhlazení šumu zachováváme ostré výrazné jasové nespojitosti - hrany. Taková metoda se ukázala být významným způsobem závislá na volbě parametrů (především se jedná o parametr rozmazání λ a časové kroky). Nalezení algoritmu, který by hodnotu parametru λ určil podle vstupních informací (struktury obrázku a pod.), je jedním ze směrů, které by se mohly dále podrobněji zkoumat.

Zjištěné výsledky ukazují, že pro různé typy znehodnocení obrazu jsou vhodné rozdílné techniky. Například pro Gaussovský šum dává numerická metoda pro nelineární parabolickou rovnici velmi podobné výsledky jako vyhlazovací filtrační masky a záleží na uživateli, jestli pro další zpracování využije obrázek s ostrými hranami a nedokonale odstraněným šumem, nebo částečně rozmazaný obrázek bez šumu - to se v případě numerického přístupu ovlivňuje volbou parametru λ . Pro šum označovaný výrazem sůl a pepř se použití numerické metody neosvědčilo, lepší výsledky poskytuje mnohem jednodušší a časově méně náročný filtr medián. Jak se ukázalo, v některých případech klasické metody fungují výrazně lépe než numerické.

Popsané numerické metody pro zkoumané parciální diferenciální rovnice jsou naprogramovány v softwaru MATLAB a přiloženy včetně testovacích obrázků na CD. Seznam všech přiložených souborů naleznete v příloze A.

Existují numerické metody, které nejen že odstraní nežádoucí šum, ale také zrestaurují hrany v rozmazaném obrázku - v nejhorším případě je vstupní obrázek rozmazaný a zároveň poškozený šumem. Zkoumání a vylepšování takových metod by bylo velkým přínosem do numerického zpracování obrazu. Kromě zde zkoumaných klasických a numerických přístupů existují dále iterační metody s regularizací, které by také mohly být předmětem zkoumání a srovnání.

V této práci jsme se zabývali převážně šedotónovými obrázky, v praxi se ale většinou setkáváme s barevnými. Zde uvedené algoritmy jsou použitelné i po zpracování barevných obrázků. Možností řešení je rozložení barvy do tří barevných vrstev (červená, zelená a modrá) a výpočet pro jednotlivé barevné složky zvlášť - tedy jedná se o třikrát opakovaný výpočet odpovídající šedotónovému obrázku. Zajímavou myšlenkou, která by se mohla aplikovat na barevné obrázky, je převod běžného barevného RGB systému do HSV modelu, kde by se redukoval šum pouze v barevné složce.

Literatura

- [1] http://cs.wikipedia.org/wiki/Elektromagnetické_spektrum.
- [2] <http://www.12on14.com/dpi/rgb-cmyk.jpg>.
- [3] http://ca.wikipedia.org/wiki/Model_de_color_HSV.
- [4] J. A. Trangenstein. *Numerical Solution of Elliptic and Parabolic Partial Differential Equations*. Lecture Notes in Computer Science. Cambridge University Press, Cambridge UK, 2013.
- [5] M. Brandner, J. Egermaier, and H. Kopincová. Numerické metody pro řešení evolučních parciálních diferenciálních rovnic. 2012.
- [6] S. K. Godunov. Reminiscences about Numerical Schemes. 2008.
- [7] A. Kratochvílová. Parciální diferenciální rovnice ve zpracování obrazu. 2007.
- [8] J.A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 1999.
- [9] J. A. Rossmanith. Conservation Laws and Applications. University of Wisconsin, Madison, USA, 2010.
- [10] M. Železný. Materiály k předmětu KKY/Zpracování digitalizovaného obrazu.
- [11] http://sprobotics.blogspot.cz/2012_02_01_archive.html.
- [12] M. Pesenson, W. Roby, and B. McCollum. Multiscale Astronomical Image Processing Based on Nonlinear Partial Differential Equations. 2008.
- [13] R. C. Gonzalez and R. E. Woods. Spatial Domain Filtering Edge Detection.
- [14] A. Belyaev. On Implicit Image Derivatives and Their Applications. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2011. <http://dx.doi.org/10.5244/C.25.72>.
- [15] B. Jůzová. Numerické metody typu upwind pro řešení parciálních diferenciálních rovnic ve více prostorových dimenzích. 2013.
- [16] A. Marquina and S. Osher. Explicit Algorithms for a New Time Dependent Model Based on Level Set Motion for Nonlinear Deblurring and Noise Removal. *SIAM J. Sci. Comput.*, 22:387–405, 1999.
- [17] V. V. Vlček. Filtrace digitálních jednorozměrných a dvojrozměrných dat inverzní Radonovou transformací. 2007.
- [18] <http://www.mathworks.com/help/matlab/ref/ode45.html>.
- [19] Z. Baxová. Numerické metody pro advekční rovnici. 2012.

- [20] <http://www.mathworks.com/help/matlab/ref/ode15s.html>.
- [21] <http://www.mathworks.com/help/comm/ref/awgn.html>.
- [22] [http://www.ece.rice.edu/~\sim\\$wakin/images/](http://www.ece.rice.edu/~\sim$wakin/images/).
- [23] [http://www.ele.uri.edu/~\sim\\$hansenj/projects/ele585/lab5/](http://www.ele.uri.edu/~\sim$hansenj/projects/ele585/lab5/).
- [24] http://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_imgproc/py_filtering/py_filtering.html#filtering.

Příloha A

Seznam příložených souborů

Zde jsou uvedeny všechny soubory přiložené na CD. U každého hlavního souboru jsou vypsány všechny pomocné soubory a testovací obrázky.

- Lineární hyperbolická rovnice (sekce 6.1)

- *hyperbolic_posun_main.m*
 - * *hyperbolic_posun_advection.m*
- *hyperbolic_posun_toky_main.m*
 - * *hyperbolic_advection.m*
 - * *hyperbolic_funF.m*
 - * *hyperbolic_funG.m*
- *hyperbolic_main.m*
 - * *hyperbolic_advection.m*
 - * *hyperbolic_funF.m*
 - * *hyperbolic_funG.m*
- *image_gray_main.m*
 - * *image_advection.m*
 - * *picture1.png*
 - * *picture2.png*
 - * *picture3.png*
- *image_rgb_main.m*
 - * *image_advection.m*
 - * *picture1rgb.png*
 - * *picture2rgb.jpg*

- Lineární parabolická rovnice (sekce 6.2)

- *parabolic_expl_main.m*
 - * *parabolic_advection.m*
 - * *parabolic_funF.m*
 - * *parabolic_funF.m*
- *parabolic_impl_main.m*
 - * *parabolic_advection.m*
 - * *parabolic_funF.m*
 - * *parabolic_funF.m*

PŘÍLOHA A. SEZNAM PŘILOŽENÝCH SOUBORŮ

- Nelineární parabolická rovnice v 1D (sekce 6.3)

- *parabolic_nelin_1D.m*

- Nelineární parabolická rovnice v 2D (sekce 6.4)

- *parabolic_nelin_2D.m*

- * *lena_snr3.bmp*

- * *lena_sum.bmp*

- * *lena_sum2.bmp*

- * *mona_lisa_sum.bmp*

- * *cameraman_sum.bmp*

- * *image1_sum.bmp*

- * *image1_sum2.bmp*

- * *image1_sum3.bmp*