

České vysoké učení technické v Praze
Fakulta jaderná a fyzikálně inženýrská

Katedra fyziky
Obor: Fyzikální inženýrství
Zaměření: Fyzika a technika termojaderné fúze



Tomografie plazmatu na tokamaku
COMPASS
Plasma tomography on the
COMPASS tokamak

BAKALÁŘSKÁ PRÁCE

Vypracoval: Michal Odstrčil
Vedoucí práce: RNDr. Jan Mlynář Ph.D.
Rok: 2010

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady uvedené v přiloženém seznamu.

Nemám závažný důvod proti užití tohoto díla ve smyslu § 60 zákona č.121/200Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne

.....
Michal Odstrčil

Poděkování

Děkuji RNDr. Janu Mlynářovi, Ph.D. za vedení mé bakalářské práce a za podnětné návrhy, které ji obohatily.

Michal Odstrčil

Název práce:

Tomografie plazmatu na tokamaku COMPASS

Autor: Michal Odstrčil

Obor: Fyzikální inženýrství

Druh práce: Bakalářská práce

Vedoucí práce: RNDr. Jan Mlynář Ph.D.
ÚFP AV ČR

Abstrakt: Tato práce seznamuje čtenáře se základními algoritmy pro tomografickou rekonstrukci s tím, že je zde detailně popsán algoritmus hledání minima Fisherovy informace a určení geometrie pro tento algoritmus. Kromě popisu byly i otestovány různé vlastnosti a vylepšení algoritmu na modelových reálných datech.

Klíčová slova: tokamak, tomografie, COMPASS, minimum Fisherovy informace

Title:

Plasma tomography on the COMPASS tokamak

Author: Michal Odstrčil

Abstract: This thesis shows basic algorithms for tomographic reconstruction. The algorithm of finding the minimum of the Fisher information is described in detail including determining the geometry. In addition to describing were also tested different properties and enhancements of the algorithm.

Key words: tokamak, tomography, COMPASS, minimum of the Fisher information

Obsah

Úvod	7
1 Analytické algoritmy v tomografii plazmatu	10
1.1 Radonova transformace	10
1.1.1 Inverzní Radonova transformace	11
1.2 Abelova transformace	12
1.3 Cormackova metoda	14
1.4 Fourier-Besselova metoda	17
2 Pixelové metody	19
2.1 Hodnotící funkce	21
2.1.1 Lineární regularizace	21
2.1.2 Minimum Fisherovy informace - MFI	22
2.1.3 Vážená druhá derivace - MVDR	23
2.1.4 Princip maximální entropie	23
2.1.5 Neuronové sítě	24
2.2 Porovnání algoritmů	24
3 Tichonovova regularizace s minimem Fisherovy informace	26
3.1 Časové rozlišení (Rapid verze)	29
4 Geometrie	31
4.1 Geometrická matice	33
4.1.1 Kruhový profil pixelů, tzv. „Abelizace“	39
4.2 Matice derivace	40
4.2.1 Derivace ve směru	40

4.2.2	Derivace podle magnetického pole	41
5	Numerické simulace	43
5.1	Porovnání algoritmů	43
5.2	Použití různé velikosti šumu	45
5.3	Systematická chyba	45
5.4	Počáteční podmínky	47
5.5	Použití jednoduchých chord	49
5.6	Podmínka nezápornosti vyzařování	49
5.7	Matice derivace podle magnetického pole	50
5.8	Optimalizace algoritmu	51
5.8.1	Řídké matice	51
5.8.2	Hledání kořene	52
5.8.3	Iterační algoritmy	54
5.8.4	Počáteční podmínky	54
6	Rekonstrukce reálných dat	55
	Závěr	57
	Seznam použitých zdrojů	60
	Přílohy	62
A	Obsah CD	62

Úvod

Termonukleární fúze představuje prakticky nevyčerpatelný zdroj energie, který by mohl v budoucnu nahradit energii z fosilních paliv a štěpných reakcí. V dnešní době ještě nejsou fúzní elektrárny potřeba, protože dokážeme pokrýt spotřebu energie z levnějších zdrojů, ale v příštím století by už mohla být situace velmi odlišná. Nepředpokládá se, že by fosilní paliva došla, ale je velmi pravděpodobné, že se několikanásobně zvýší jejich cena a tedy i výsledná cena elektrické energie. Také uranu ^{235}U , který se používá v jaderných elektrárnách 1-3. generace, bude méně a tedy i cena elektrické energie vyrobené v klasických jaderných elektrárnách se zvýší. Zbývají sice ještě obnovitelné zdroje, ale na pouze obnovitelné zdroje se lidstvo nemůže spoléhat. Z toho vyplývá, že jaderná fúze spolu s jadernými elektrárnami 4. generace budou s velkou pravděpodobností významný zdroj elektřiny pro příští století.

Aby mohla fúze být použita jako zdroj energie, tak je nutné vyřešit ještě mnoho technických problémů. Přes 50 let se vědci snažili zahřát a stlačit plazma na takovou teplotu, aby se jaderná fúze stala významným zdrojem tepla, čehož se podařilo úspěšně dosáhnout v tokamaku JET v Anglii. Další krok ve vývoji je zaměřený na dosažení stavu vyrovnaní, kdy je fúzní výkon roven výkonu, který je potřebný pro dodatečné ohřívání plazmatu. Toho se plánuje dosáhnout a dokonce desetkrát překročit v tokamaku ITER, který se staví ve Francii. Třetí krok bude fúzní elektrárna DEMO, která by měla ověřit ekonomickou konkurenceschopnost jaderné fúze.

Dnešní výzkum je zaměřený na dva hlavní problémy. Najít materiály, ze kterých by bylo možné postavit spolehlivý fúzní reaktor, a maximalizovat dobu udržení plazmatu. Doba udržení lze zvýšit regulací provozu fúzního reaktoru s cílem udržení optimálních parametrů plazmatu po co nejdelší čas. A právě z tohoto důvodu je potřeba chování plazmatu studovat a měřit.

Avšak měřit vlastnosti plazmatu v tokamaku je poměrně obtížné. Jednak je plazma velmi horké, takže snadno poškodí senzory, které by měly měřit jeho vlastnosti, na druhou stranu má nízkou hustotu a tedy i tepelnou kapacitu, díky čemuž pokusy o fyzické změření vlastností mohou plazma velmi ovlivnit. Přestože existuje velké množství způsobů, jak diagnostikovat vlastnosti plazmatu, většina metod je nepřímá a tedy i nepřesná. Často se měří vlastnosti pouze v poloidálním řezu tokamakem a předpokládá se, že ze silné symetrie podle hlavní osy budou hodnoty v ostatních částech tokamaku podobné.

Běžně používaná diagnostika je měření magnetických vlastností plazmatu. Například pomocí magnetické rekonstrukce lze přibližně určit polohu plazmatu v tokamaku. Další možnost je měření rozptylu laseru v plazmatu a z odraženého laserového sig-

nálu určovat vlastnosti plazmatu nebo měřit vyzařování svazků neutrálních částic vstříkovaných do plazmatu.

Vodíkové plazma v tokamaku má tak vysokou teplotu, že je plně disociované, a proto nevyzařuje světlo přechody elektronů mezi hladinami jako klasická látka. Vyzařování pomocí přechodů nastává pouze od příměsí v plazmatu a u tokamaku COMPASS je nejintenzivnější v oblasti UV záření. Velká část záření plazmatu vzniká jako brzdné záření při srážkách nabitých částic, tedy hlavně srážkách elektronů, protože elektrony mají mnohem vyšší rychlosti a jejich srážky jsou intenzivnější. Srážky ve středu plazmatu jsou tak energetické, že už vzniká měkké rentgenové záření. Celkově se dá říci, že z prostorového rozložení vyzařování plazmatu lze odhadnout rychlost částic a tedy jejich teplotu v různých částech tokamaku. Když se navíc získá i časový průběh, tak lze studovat transportní jevy, unikání částic z magnetických ploch nebo turbulence.

Tomografie

Slovo tomografie pochází z řeckého slova $\tau\omicron\mu\omicron\varsigma$ +grafie tedy „zobrazování řezu“. Základním principem tomografie je snaha převést obraz získaný na okraji předmětu na obraz uvnitř bez jeho fyzického porušení.

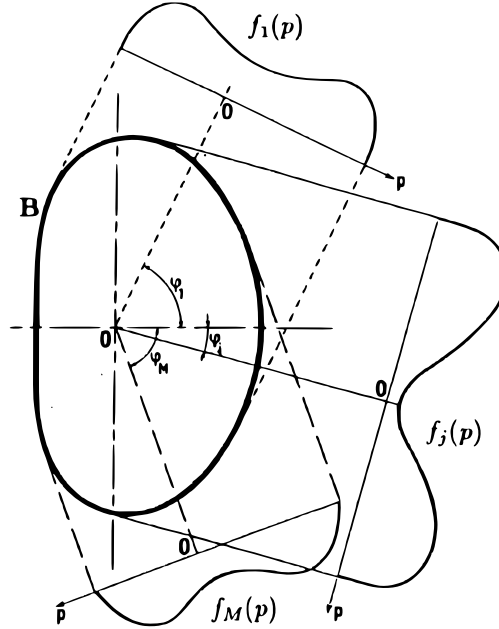
Tomografie je jedním z prostředků na diagnostiku plazmatu, ale kromě fyziky plazmatu má široké využití v mnoha vědních oborech v medicíně, technice, archeologii a mnoha dalších. Většinou jsou známy hodnoty projekcí získané na okraji měřeného objektu, např. ve fyzice plazmatu výkon vyzářený v určitém směru, v lékařství naopak intenzita rentgenového záření, které prošlo v daném směru. A z těchto projekcí je potřeba získat co nejpřesnější řez objektem. V tokamacích se dá tomografie použít pro pozorování různých transportních jevů a nestabilit. Běžně se pro určování intenzity záření plazmatu určování používají rychlá bolometrická měření (měření celkového vyzářeného výkonu) a měření měkkého rentgenového záření (SXR).

Z důvodu, že z integrálních hodnot je tomografie schopna určit lokální hodnoty, je tomografie v diagnostice plazmatu těžko zastupitelná ostatními diagnostickými metodami. Naopak znalost dalších informací z ostatních metod lze využít ke zpřesnění *tomografické rekonstrukce*.

Tomografická rekonstrukce je matematický proces, kdy z funkce $f(\vec{p})$ získané na okraji získává zpětně funkce $g(\vec{x})$ uvnitř. Velký problém tomografie je, že tato úloha je špatně podmíněná, to znamená, že i malá chyba v měření $f(\vec{p})$ se může výrazně projevit v rekonstruované funkci $g(\vec{x})$. V tomografii není důležité, jestli funkce $g(\vec{x})$ vyjadřuje hustotu vyzářeného výkonu v bodě $\vec{x}$ jako u plazmatu nebo naopak hustotu pohlcování rentgenového záření jako v medicíně. Liší se jen některé drobnosti týkající se sestavování funkce $g(\vec{x})$, například při rentgenové tomografii v medicíně se musí započítat exponenciální útlum záření, naopak v plazmatu je vhodné započíst vliv prostorového úhlu, pod kterým vidí plazma detektor a mnoho dalších jevů. Asi největší rozdíl mezi tomografií plazmatu a tomografií v medicíně je ale v množství údajů, například tomografii plazmatu je to počet úhlů a směrů projekcí, ze kterých se tomografická rekonstrukce skládá. Zatímco v lékařství je to řádově 10^5 údajů, ve

kterých je známa hodnota funkce $f(\vec{p})$, u plazmatu je to z důvodu nedostatku místa limitované řádově na 10^2 údajů [3].

Další výhodou tomografie plazmatu by v budoucnu mohla být rekonstrukce v reálném čase mimo jiné i díky stále rychlejším počítačům. Pomocí toho by bylo možné mimo jiné odhalovat nestability a korigovat je dříve než se plně projeví nebo ověřovat měření polohy plazmatu určené z magnetické rekonstrukce.



Obrázek 1: Schéma tomografické rekonstrukce pro M úhlů, převzato z [4]

Kapitola 1

Analytické algoritmy v tomografii plazmatu

1.1 Radonova transformace

Radonova transformace je nejzákladnější případ analytického řešení tomografické rekonstrukce. Transformace je definována vztahem mezi obrazem (projekcí) f a vzorem (zdrojem) g takto

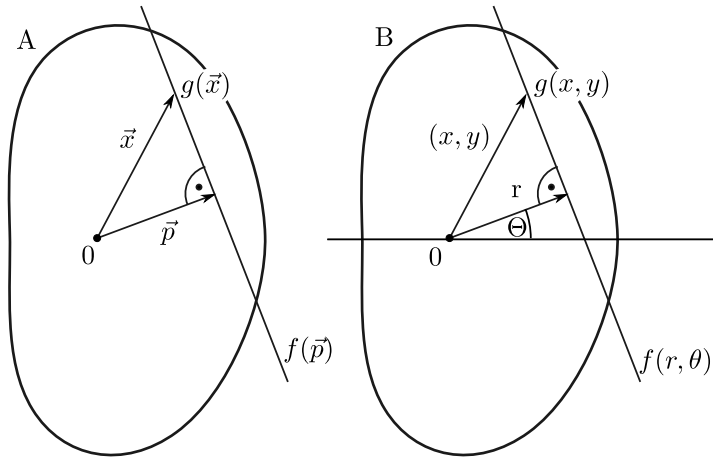
$$\hat{R}\{g(\vec{x})\} \equiv f(\vec{p}) = \int_{\mathbb{R}^N} g(\vec{x}) \delta(\vec{x} \cdot \vec{n} - p) d^N x \quad (1.1)$$

kde $\hat{R}\{g(\vec{x})\}$ značí Radonovu transformaci, $g(\vec{x})$ je N -dimenzionální funkce, $f(\vec{p})$ je její průmět a $\vec{p}$ probíhá po $N-1$ dimenzionální ploše, přes kterou se pozoruje funkce g , $p = |\vec{p}|$ a $\vec{n} = \vec{p}/|\vec{p}|$. Pro případ dvourozměrného řezu lze rovnici zjednodušit

$$f(r, \theta) = \int_{\mathbb{R}^2} g(x, y) \delta(x \cos \theta + y \sin \theta - r) dx dy \quad (1.2)$$

Tato rovnice popisuje stejně jako předchozí integraci funkce g po přímce, ale v tomto případě značí r kolmou vzdálenost přímky od středu rekonstrukce a θ je úhel, který přímka svírá s osou x . Použité značení je vykreslené na obrázku (1.1).

Cílem je určit původní funkci g , k tomu nám stačí v ideálním případě jenom získat inverzní transformaci $\hat{R}^{-1}\{f(\vec{p})\} = g(\vec{x})$.



Obrázek 1.1: Schéma transformace přímek, obrázek A vektorové značení, obrázek B polární značení

1.1.1 Inverzní Radonova transformace - centrální řezový te- orém

Nechť $\hat{F}$ je operátor n -rozměrné Fourierovy transformace definovaný jako

$$\tilde{g}(\vec{\rho}) = \hat{F}\{g(\vec{x})\} = \int_{\mathbb{R}^N} g(\vec{x}) \exp(-2\pi i \vec{x} \cdot \vec{\rho}) d^N x \quad (1.3)$$

když se upraví $\hat{F}$ pomocí δ -funkce, tak se dostane vztah

$$\tilde{g}(\vec{\rho}) = \hat{F}\{g(\vec{x})\} = \int_{\mathbb{R}} dt \int_{\mathbb{R}^N} g(\vec{x}) \exp(-2\pi i t) \delta(t - \vec{x} \cdot \vec{\rho}) d^N x \quad (1.4)$$

kde exponenciála závisí pouze na parametru t , protože všude mimo je nulovaná δ -funkcí. Nyní se zavede nový parametr p substitucí $t = p\rho$, kde $\rho = \|\vec{\rho}\|$, potom lze integrál rozdělit

$$\tilde{g}(\vec{\rho}) = \rho \int_{\mathbb{R}} \exp(-2\pi i p \rho) dp \int g(\vec{x}) \delta(p - \rho \vec{x} \cdot \vec{\rho}) d^N x \quad (1.5)$$

kde druhý integrál je mírně upravená Radonova transformace a první integrál je naopak jednorozměrná Fourierova transformace, proto po úpravě vychází

$$\tilde{g}(\vec{\rho}) = \rho \int_{\mathbb{R}} \exp(-2\pi i p \rho) \hat{R}\{g(\vec{x})\} dp = \hat{F}_1 \hat{R}\{g(\vec{x})\} \quad (1.6)$$

A to je tvrzení *centrálního řezového teoremu*:

$$\hat{F} = \hat{F}_1 \hat{R} \quad (1.7)$$

Z toho lze přímo vyjádřit vzorec pro inverzní Radonovu transformaci,

$$\hat{R}^{-1} = \hat{F}^{-1} \hat{F}_1 \quad (1.8)$$

čímž se ukazuje, že Radonova transformace má inverzní transformaci na vhodně zvolených prostorech funkcí. Pomocí tohoto vzorce lze získat původní funkce $g(x, y)$ celkem přesně i rychle [5], když se použije algoritmus rychlé Fourierovy transformace.

Pokud se rekonstrukce omezí pouze na $\mathbb{R}^2$, tak lze z předchozího vzorce odvodit přímo vztah pro inverzní transformaci [4, 6]

$$g(x, y) = \frac{1}{4\pi^2} \int_0^{2\pi} \int_{\mathbb{R}} \frac{f'(r, \theta)}{x \cos \theta + y \sin \theta - r} d\theta dr$$

Problémem těchto řešení je potřeba přesné znalosti hodnoty funkce $f(r, \theta)$ ve všech nebo alespoň velmi mnoha bodech. Pokud by byla známa hodnota funkce $f(r, \theta)$ v nekonečně mnoha bodech, potom by šla funkce $g(x, y)$ určit přesně, což bohužel tomografie plazmatu nesplňuje. Pokud je měření řídké, tak sice lze hodnoty proložit a interpolovat, ale tato transformace není moc odolná proti chybám a výsledná rekonstrukce může být velmi poškozená nebo mohou vznikat problémy se singularitami. Z toho důvodu se tato metoda v tomografii nepoužívá. Je na ní ale založená metoda Filtrované zpětné projekce, která se používá v lékařské tomografii [6].

1.2 Abelova transformace

Abelova transformace předpokládá úplnou středovou symetrii rekonstruované dvourozměrné funkce. To je velmi silný předpoklad, proto se používá vylepšená verze tzv. asymetrická Abelova transformace [8], která je založená na klasické, ale předpokládá zrcadlovou symetrii. Předpoklad středové symetrie převede dvourozměrný problém na jednorozměrný.

Zmíněné požadavky na symetrii jsou v plazmatu často splněny, díky poměrně velké poloidální symetrii. Jediný další údaj, který je pro rekonstrukci potřeba, je poloha středu plazmatu. Ta se dá například určit pomocí magnetické rekonstrukce. Nejzajímavější věc na Abelově metodě je to, že na to jak je jednoduchá, tak její výsledky mohou být poměrně přesné. Druhá výhoda této metody je v tom, že k rekonstrukci stačí pouze jeden detektor.

Nejdříve je třeba odvodit vztah pro symetrickou Abelovu transformaci. Základní vztah, ze kterého se vychází, je

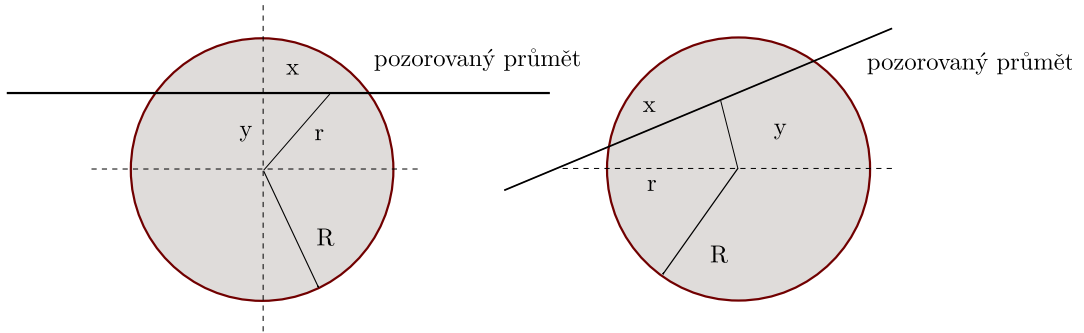
$$f(y) = \int_{\mathbb{R}} g(x, y) dx \quad (1.9)$$

kde jako v předchozím případě $g(x, y)$ značí kruhově symetrickou hustotu rozdělení měřené veličiny a $f(y)$ je projekce, v tomto případě se používá jen pohled z y osy, schéma je na obrázku (1.2). Pokud se využije radiální symetrie, tak po substituci $x^2 + y^2 = r^2$ se dostane

$$f(y) = 2 \int_0^\infty g(\sqrt{x^2 + y^2}) dx = 2 \int_y^\infty \frac{g(r)r}{\sqrt{r^2 - y^2}} dr \quad (1.10)$$

navíc díky radiální symetrii je možné případ, kdy f je závislé na y snadno upravit na vějířovitý případ, kdy se mění úhel α a vzdálenost D zůstává konstantní (viz obr. 1.2). Pak se předchozí vzorec upraví

$$f(\alpha) = 2 \int_y^\infty \frac{g(r)r}{\sqrt{r^2 - (D \sin \alpha)^2}} dr \quad (1.11)$$



Obrázek 1.2: Schéma Abelovy transformace, vlevo je pohled z osy y a vpravo je vějířovitá transformace

Řešení inverzní Abelovy transformace je jednodušší než Radonova transformace. Pro inverzní transformaci platí vztah

$$g(r) = -\frac{1}{\pi} \int_r^\infty \frac{df}{dy} \frac{dy}{\sqrt{y^2 - r^2}} = -\frac{1}{\pi} \int_{\arcsin(r/D)}^{\pi/2} \frac{df}{d\alpha} \frac{d\alpha}{\sqrt{(D \sin(\alpha))^2 - r^2}} \quad (1.12)$$

To, že jde opravdu o inverzní vztah lze snadno ověřit. Je potřeba předpokládat, že g a $\frac{dg}{dr}$ jsou hladké a klesají v nekonečno dost rychle k nule, což u plazmatu určitě platí, mimo tokamak jsou totiž nulové. Nejdříve se funkce (1.10) integruje per-partes

$$f(y) = -2 \int_y^\infty g'(r) \sqrt{r^2 - y^2} dr$$

potom se obě strany zderivují podle y , zde se využije předpoklad konvergence integrálu a spojitosti derivace

$$f'(y) = 2y \int_y^\infty \frac{g'(r)}{\sqrt{r^2 - y^2}} dr$$

nyní se dosadí $f'(y)$ do inverzní transformace (1.12)

$$g(r) = -\frac{1}{\pi} \int_r^\infty \frac{f'(y)}{\sqrt{y^2 - r^2}} dy = \int_r^\infty \int_y^\infty \frac{-2y g'(s)}{\pi \sqrt{(y^2 - r^2)(s^2 - y^2)}} ds dy =$$

dále podle Fubiniho věty lze nyní zaměnit integrály

$$\begin{aligned} \int_r^\infty \int_r^s \frac{-2y}{\pi \sqrt{(y^2 - r^2)(s^2 - y^2)}} dy g'(s) ds &= \int_r^\infty -\frac{1}{\pi} \left[\operatorname{arctg} \left(\frac{r^2 + s^2 - 2y^2}{2\sqrt{s^2 - y^2}\sqrt{y^2 - r^2}} \right) \right]_r^s g'(s) ds = \\ &= \int_r^\infty -g'(s) ds = g(r). \end{aligned}$$

čímž, je tvrzení dokázáno.

Derivace funkce f v inverzním vztahu se získává tak, že se naměřená data proloží vhodnou funkcí [7], například kombinací polynomů a Gaussovy funkce, čímž se zároveň zvýší odolnost metody vůči šumu.

Tuto metodu je možné upravit z předpokladu kruhové symetrie na mírnější předpoklad a to je zrcadlová symetrie zavedením váhové funkce $w(y)$. Pro rekonstrukci se v tomto případě použije symetrická část funkce f a antisymetrická část se použije pro váhovou funkci, tzn.

$$f_s(y) = (f(y) + f(-y))/2 \quad f_a(y) = (f(y) - f(-y))/2$$

a váhová funkce je pak rovna $w(y) = 1 + f_a(y)/f_s(y)$. Toto je příklad pro jeden detektor, pro více detektorů by se to řešilo obdobně. Výsledná rekonstrukce se získá ze symetrické Abelovy metody aplikované na symetrickou část. Výsledek se následně vynásobí váhovou funkcí.

$$g(y) = w(y)g_s(y)$$

Podrobněji je tento způsob zpracován v článku [8].

1.3 Cormackova metoda

Metody, které zde byly dosud zmíněny, se pokoušely vyřešit inverzi přímo hledáním funkce $g(x, y)$ tak, aby odpovídala řešení rovnice

$$f(\vec{p}) = \int g(\vec{x}) \delta(\vec{x} \cdot \vec{n} - p) d\vec{x}$$

Cormackova metoda hledá rozklad funkce $g(\vec{x})$ do speciálních polynomů v poloměru a do Fourierovy řady ve směru úhlu, tedy do ortogonálních funkcí, čímž je zaručena jednoznačnost řešení. Výhodou této metody je, že převede řešení integrální rovnice na nekonečnou soustavu lineárních rovnic. A to je šikovné, protože polynomy a lineární rovnice se na počítači řeší jednodušeji než integrály a derivace, ale pořád je to špatně podmíněná úloha, jak bylo zmíněno v úvodu.

Nejdříve se funkce g a f rozvinou do Fourierovy řady jako

$$g(r, \theta) = \sum_{m=0}^{\infty} [g_m^c(r) \cos(m\theta) + g_m^s(r) \sin(m\theta)]$$

$$f(p, \phi) = \sum_{m=0}^{\infty} [f_m^c(p) \cos(m\phi) + f_m^s(p) \sin(m\phi)]$$

Dále je potřeba určit polynomy $g_m^{c,s}$ a $f_m^{c,s}$. Pokud platí, že $0 < p < 1$, tak lze jako radiální části rozvoje $f_m^c(p), g_m^s(r)$ použít členy [4, 9]

$$f_m^{(c,s)}(p) = 2\sqrt{1-p^2} \sum_{l=0}^{\infty} a_{lm}^{(c,s)} U_m^l(p) \quad (1.13)$$

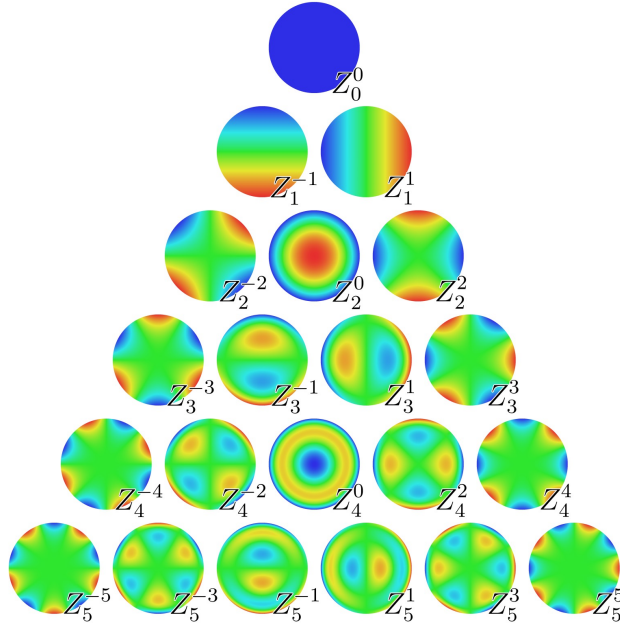
$$g_m^{(c,s)}(r) = \sum_{l=0}^{\infty} a_{lm}^{(c,s)} (m+2l+1) Z_m^l(r) \quad (1.14)$$

kde $a_{lm}^{(c,s)}$ jsou vhodně zvolené konstanty, $U_m(p)$ je Čebyševův polynom druhého druhu

$$U_m^l(p) = \frac{\sin[(m+2l+1) \arccos(p)]}{\sqrt{1-p^2}}$$

a $R_m^l(r)$ jsou radiální části Zernikových polynomů

$$R_m^l(r) = \sum_{s=0}^l \frac{(-1)^s (m+2l-s)! r^{m+2l-2s}}{s! (m+l-s)! (l-s)!}$$



Obrázek 1.3: Zernikovy polynomy nejnižších řádů, převzato z Wikipedie

Tímto se převedly špatně určené integrální rovnice¹ na rovnice polynomů, které mohou být přeurčené [3]. Přeurčené rovnice se většinou řeší metodou nejmenších čtverců. Koeficienty $a_{lm}^{(c,s)}$, které jsou pro obě funkce stejné, se získaly metodou nejmenších čtverců z rovnice (1.13), díky čemuž je metoda relativně odolná proti šumu. Když jsou známy koeficienty $a_{lm}^{(c,s)}$, tak už není problém spočítat hledanou funkci g . Tato kombinace „sdružených polynomů“ není jediná, která se dá použít. V minulosti bylo publikováno mnoho dalších, proto je potřeba vybrat nejvhodnější dvojici polynomů podle druhu úlohy, na kterou se použijí.

Do Cormackovy metody se ještě musí započítat několik dalších problémů, které při rekonstrukci nastávají. Hlavní problém je, že existuje pouze konečný počet známých hodnot průmětů $f(p, \phi)$. Protože nejdůležitější členy v rozvoji jsou členy na začátku, tak se pro konečný počet hodnot polynomy vyšších řádů zanedbají a vztahy pro rozvoj přejdou do tvaru

$$f^{(c,s)}(p) = \sum_{m=0}^{M-1} \sum_{l=0}^{L-1} a_{lm}^{(c,s)} \sin[(m+2l+1) \arccos(p)]$$

kde M je maximální počet nezávislých pohledů v prostoru s poloměrem p a L je počet chord v jednom detektoru. Pokud není v detektorech stejný počet senzorů, tak se číslo L musí určit experimentálně. Toto je jenom jeden příklad, jak je možné přejít ke konečnému počtu měřených průmětů. Obecně lze použít jakékoliv přiřazení [4], které definuje jednoznačný vztah mezi $k \in \{1, \dots, Q\}$ a indexy l, m , kde Q je celkový počet známých průmětů $f(p, \phi)$ (tzn. $Q = L \cdot M$). Potom předchozí rovnice přejde na tvar

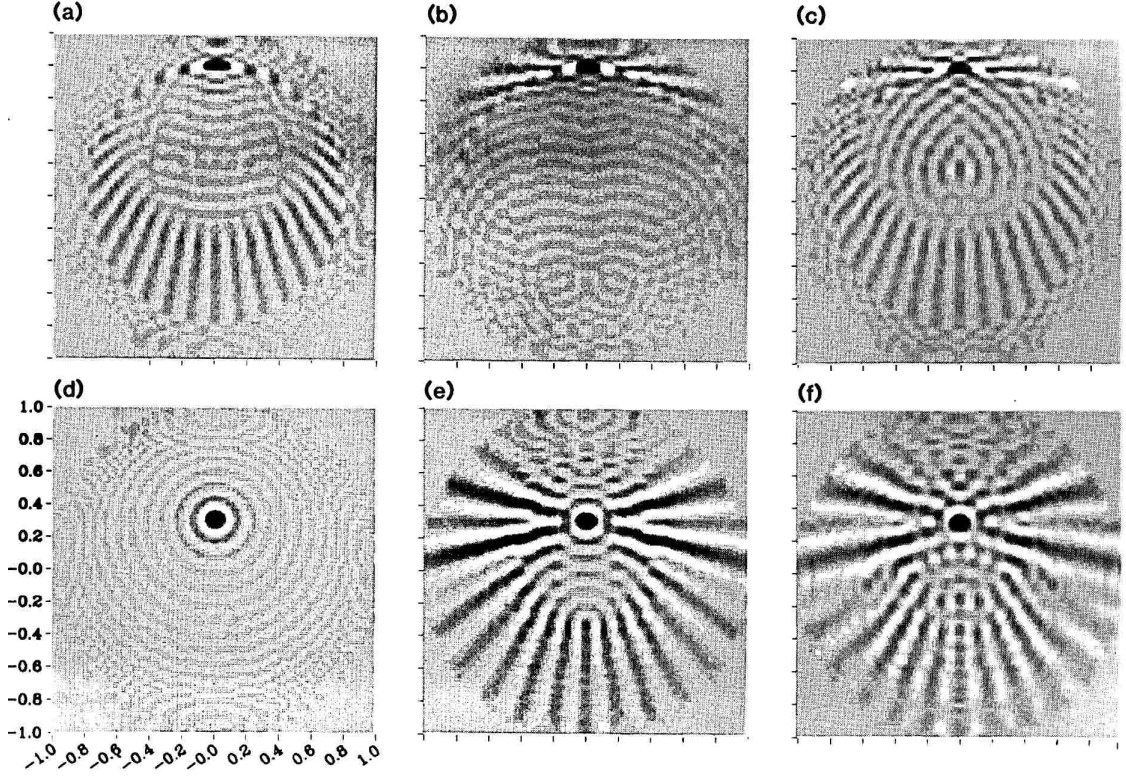
$$f^{(c,s)}(p) = \sum_{k=1}^Q a_k^{(c,s)} \sin[(m+2l+1) \arccos(p)]$$

Tím, že se odstraní horní členy řad, kterými se funkce rozvíjely, tak se objeví problém, který souvisí s konečnou délkou polynomů a to je aliasing (obr. 1.4). Obecně aliasing nastává, když je vzorkovací frekvence měřicího přístroje moc nízká oproti frekvenci dat. Obvykle se podle Shannonova teorému požaduje alespoň dvakrát vyšší snímkovaná frekvence než je frekvence vyskytující se ve funkci f . V tomto případě se spíše místo frekvence myslí charakteristická délka fluktuací

Zajímavý důsledek aliasingu je fakt, že čím lepší bude rozlišení v poloměru p tím horší bude homogenita rekonstruované funkce $g(x, y)$ a naopak, podrobněji v článku [4]. Co se myslí rozlišením v poloměru p je vidět na (obr. 1.5), kde každý jednotlivý bod reprezentuje data z jednoho senzoru. V ideálním případě je vhodné mít rovnoměrné pokrytí celého tokamaku.

Poslední velká nevýhoda je předpoklad přesné znalosti středu plazmatu. Tento problém je důkladně prostudován v práci [7], ale celkové lze říct, že pokud je rozdíl

¹tím, že je úloha špatně určená je myšleno, že počet neznámých je vyšší než počet rovnic, pro přeurčené rovnice je to naopak



Obrázek 1.4: Ukázka aliasingu vznikajícího při Cormackově rekonstrukci, v obrázcích (a-c) je zdroj umístěný v bodě 0,8 a na obrázcích (d-f) je v bodě 0,3. Na obrázcích vlevo je pouze azimutální aliasing, uprostřed je pouze radiální a vpravo je kombinace obou, převzato z [10]

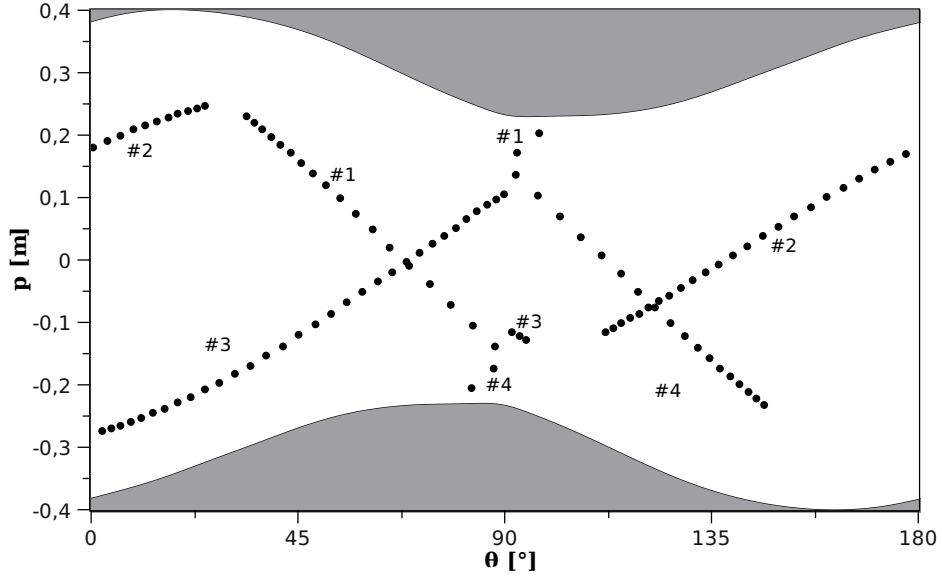
středu plazmatu a středu rekonstrukce větší, tak rekonstruovaná funkce může ztratit jakoukoli vypovídací hodnotu.

1.4 Fourier-Besselova metoda

Cormackova metoda se sice v tomografii plazmatu používá, ale protože trpí určitými problémy a nestabilitami, které jsou způsobeny použitím Zernikových polynomů, tak se často používají jiné algoritmy, které jsou na ní založené. Asi nejznámější je Fourier-Besselova metoda. Tato metoda rozkládá funkce f, g do Fourierovy řady v úhlu, stejně jako Cormackova metoda, ale místo polynomů v poloměru se používají Besselovy funkce prvního druhu [9, 11, 12]. Tvar rozvoje je podobný jako u Cormackovy metody.

$$g(r, \theta) = \sum_{m=0}^{\infty} [g_m^c(r) \cos(m\theta) + g_m^s(r) \sin(m\theta)]$$

$$f(p, \phi) = \sum_{m=0}^{\infty} [f_m^c(p) \cos(m\phi) + f_m^s(p) \sin(m\phi)]$$



Obrázek 1.5: Graf pokrytí tokamaku COMPASS pomocí SXR tomografického systému, graf vznikne po převedení chord z kartézských souřadnic do polárních (schéma 1.1), jednotlivá čísla označují různé senzory

kde ve směru poloměru se použije rozvoj pomocí Besselovy funkce

$$g_m^{(c,s)}(r) = \sum_{l=0}^{\infty} a_{ml}^{(c,s)} J_m(x_{ml}r)$$

$$f_m^{(c,s)}(p) = \sum_{l=0}^{\infty} a_{ml}^{(c,s)} f_{ml}(p)$$

$$f_{ml}(p) = 2\sqrt{1-p^2} J'_m(x_{ml}) \sum_{n=0, m \neq n}^{\infty} \delta_n J_n(x_{ml}) \sin(n\pi/2 - x_{ml}p) \left(\frac{U_{m+n-1}(p)}{m+n} + \frac{U_{m-n-1}(p)}{m-n} \right)$$

x_{ml} je $(l+1)$ -ní kořen Besselovy funkce $J_m(x)$, $\delta_0 = 1/2$, $\delta_{n \neq 0} = 1$ a $U_m(p)$ je Čebyševův polynom druhého druhu

$$U_m(p) = \frac{\sin[(m+l) \arccos(p)]}{\sqrt{1-p^2}}$$

Sumy se musí stejně jako v Cormackově metodě přizpůsobit konečnému počtu změřených dat. Hlavní výhoda této metody oproti Cormackově je odstranění velkých nepřírodných obrazů na okrajích rekonstrukce. Ale stále zůstávají nevýhody jako předpoklad kruhového řezu plazmatem, nutnost znalosti polohy středu plazmatu a složitost implementace dalších a-priori znalostí.

Kapitola 2

Pixelové metody

Pixelové metody mají základní matematický princip podobný jako předchozí metody, tedy rozkládají funkci $f(r, \theta)$ do ortogonálních funkcí. Zatímco Radonova transformace rozložila obraz do spojitě Fourierovy transformace, což je ortogonální báze, Cormackova metoda rozložila funkce f do Fourierovy řady v úhlu a polynomů v poloměru, tedy taky do ortogonálních bází, pixelové metody rozloží funkci $g(x, y)$ jednoduše do pixelů g_j , které nemají společný průnik a jsou tedy také ortogonální.

Výhodou pixelových metod je, že se už od počátku počítá s konečným počtem známých bodů funkce $f(r, \theta)$ a nevznikají tedy takové problémy jako aliasing.

Základní rovnici tomografie (1.2) se zde upraví

$$f_i = \int T_i(x, y)g(x, y) dx dy \quad i \in \{1, \dots, L\}$$

kde f_i odpovídá hodnotě původní funkce $f(r, \theta)$ v bodě číslo i , L je počet známých průmětů tzn. hodnot funkce f a funkce $T_i(x, y)$ je geometrický faktor, který je podrobněji popsán v kapitole 4. V této kapitole se bude předpokládat, že $T_i(x, y)$, f_i jsou známé funkce, které závisí pouze na geometrii tokamaku. Potom se funkce $g(x, y)$ rozloží do diskretních pixelů pomocí funkce $b_j(x, y)$

$$g(x, y) = \sum_{j=1}^N g_j b_j(x, y)$$

nakonec rovnice (1.2) přejde na soustavu lineárních rovnic

$$f_i = \sum_{j=1}^N T_{ij} g_j \quad i \in \{1, \dots, L\} \quad (2.1)$$

kde g_j je hodnota j -tého pixelu, N je počet pixelů, ten se volí podle požadovaného rozlišení a T_{ij} je vliv j -tého pixelu na i -tý detektor. Pixely sice po rekonstrukci tvoří jakousi matici nebo mříž, obecně může být rozložení pixelů libovolné a nepravidelné. V této práci se používá pouze pravidelná obdélníková síť čtvercových pixelů, která po rozložení tvoří vektor g .

Tímto se převedl problém řešení integrálů na řešení soustavy lineárních rovnic. V ideálním případě by stačilo pouze invertovat matici T_{ij} a získat vektor g_j , ale to nefunguje. Matice T_{ij} není obecně čtvercová, protože počet pixelů g_j (neznámých) je zpravidla vyšší než počet hodnot f_i , tedy úloha je špatně určená.

Na tento druh úloh lze výhodně použít metodu nejmenších čtverců, kdy pro nalezení pseudoinverzní matice [13] se hledá minimum funkcionálu

$$U^2 = \sum_{i=1}^L \left(f_i - \sum_{j=1}^N T_{ij} g_j \right)^2 \quad (2.2)$$

Minimum výrazu (2.2) lze vždy nalézt a řešení je nekonečně mnoho, protože pokud nejsou dány nějaké další podmínky, pak počet parametrů (g_j) je vyšší než počet hodnot (f_i). Pokud by se našlo takovéto řešení bez dalších podmínek, pak se nazývá „over-fitting“. Takovéto řešení fituje veškerý šum, který se v datech nachází a nemá moc vysokou předpovědní hodnotu.

Protože změřené hodnoty nejsou dokonale přesné, ale obsahují náhodné chyby s normálním rozdělením $N(0, \sigma_i^2)$, tak je potřeba rovnici (2.2) mírně upravit, aby mohly být započteny chyby různé přesnosti jednotlivých hodnot f_i . Pokud se použije obecný vztah pro střední kvadratickou odchylku

$$\sigma = \sqrt{\sum_{i=1}^N \frac{(y_i - F_i(x_i, \vec{g}))^2}{N - k}} \quad (2.3)$$

kde y_i jsou naměřené hodnoty, $F_i(x_i, \vec{g})$ jsou hodnoty fitované funkce v těchto bodech, N je počet měření a k je počet volných (lineárně závislých) parametrů funkce F_i . Tak je zřejmé, že pro $N \leq k$ ani nelze určit chybu, protože existuje vždy přesné řešení (již zmíněný over-fitting). Pokud se dodá nějaká další podmínka (například hladkost) na parametry g_j , tak bude množství volných parametrů $k \ll N$ a k je možné zanedbat. Předchozí rovnici lze dosazením z rovnice (2.2) upravit na funkcionál

$$S^2 = \sum_{i=1}^L \left(\frac{f_i - \sum_{j=1}^N T_{ij} g_j}{\sigma_i} \right)^2 \quad (2.4)$$

kde σ_i jsou chyby jednotlivých hodnot f_i a L je celkový počet hodnot f_i .

Otázka ovšem je, jak moc přesně musí platit rovnice $f_i = \sum_{j=1}^N T_{ij} g_j$, aby v rámci náhodné chyby bylo možné říct, že rovnost platí. S^2 je rovno součtu kvadrátů L členů, které mají podle Centrální limitní věty [14] rozdělení $N(0,1)$ a přesně takto je definováno Pearsonovo rozdělení χ^2 s L stupni volnosti. Proto pokud $S^2 > \chi^2_{1-\alpha}(L)$, potom lze vyloučit rovnost $f_i = \sum_{j=1}^N T_{ij} g_j$ s $\alpha \cdot 100\%$ pravděpodobností chyby. Například pro 100 senzorů a 90% pravděpodobnost musí být S^2 menší než 118, naopak by mělo být menší než ~ 90 , protože jinak nastává overfitting. V dalším textu se bude předpokládat přímo rovnost $S^2 = L$ a definuje se funkcionál χ^2

$$\chi^2 = S^2 / L = 1 \quad (2.5)$$

2.1 Hodnotící funkce

Metodou nejmenších čtverců se omezí množina řešení na $N - L$ rozměrnou varietu. Ovšem k nalezení správného řešení je potřeba přidat ještě nějaké omezující podmínky. Jako častá omezující podmínka se bere hladkost funkce, další možnost je například volit řešení s nejvyšší entropií, MLE tedy řešení s maximální pravděpodobností, možností je mnoho. Řešení se potom hledá jako extrém hodnotící funkce na varietě vymezené rovnicí $\chi^2 = 1$. Tento matematický problém se běžně řeší pomocí variačního počtu Lagrangeovými multiplikátory s hodnotící funkcí H

$$\Lambda = H - \lambda(\chi^2 - 1) \quad (2.6)$$

U pixelových metod se Lagrangeův multiplikátor umísťuje před hodnotící funkci, zřejmě protože χ^2 se používá jako podmínka ve všech typech pixelových metod, zatímco hodnotící funkce H se může lišit a omezujících podmínek může být obecně i víc. Výsledný obecný tvar rovnice je tedy

$$\Lambda = \chi^2 + \sum_i \lambda_i H_i \quad (2.7)$$

Řešení se hledá tak, aby byla splněná okrajová podmínka $\chi^2 = 1$ a zároveň, aby všechny derivace $\frac{\partial \Lambda}{\partial g_i}$ byly nulové. Počítá se s tím, že hodnoty vektoru g_i jsou jediné proměnné.

2.1.1 Lineární regularizace

V dalším kroku je potřeba nalézt nejvhodnější hodnotící funkci. Často používané jsou hodnotící funkce, které by upřednostňovaly hladkost a základním typem těchto funkcí jsou *lineární regularizace*. Základní tvar těchto hodnotících funkcí je tento

$$\mathcal{R} = \|g^{(k)}\|^2$$

pro nejjednodušší případ $k = 0$ je lineární regularizace nultého řádu¹ $\mathcal{R} = \|g\|^2 = g^T g$. Tato regularizace upřednostňuje co nejnižší a nejplošší funkce, což není vhodné pro tomografii plazmatu. Výsledná funkce má píky nižší, než jsou správné hodnoty.

Další případ $k = 1$ je regularizace prvního řádu $\mathcal{R} = \|\nabla g\|^2 = \|\partial_x g\|^2 + \|\partial_y g\|^2$, kde $\partial_x g$ je derivace ve směru² x . Pokud se dosadí za normu a difference se nahradí maticí derivace (podrobněji v sekci 4.2), tak je $\|\partial_x g\|^2 = (B_x g)^T (B_x g) = g^T B_x^T B_x g$. Regularizace prvního řádu tedy preferuje co nejnižší velikost derivace v rámci omezení daného rovnicí $\chi^2 = 1$. Důsledek je, že výsledná funkce nemusí být hladká, ale bude

¹normou $\|g\|$ se v této práci bude vždy myslet euklidovská norma $\sqrt{\sum g_i^2} = \sqrt{g^T g}$

²protože g není funkce, ale vektor, tak se místo derivace použijí konečné difference $\partial_y g_i \equiv g_i - g_{i-1}$ a pro směr x bude $\partial_x g_i \equiv g_i - g_{i-N}$, kde N je počet pixelů na jednom řádku, podrobněji v sekci 4.2

mít minimální růst. Z toho také plyne, že rekonstruovaná funkce bude mít píky nižší než testovací funkce nebo měření data.

Teprve regularizace druhého řádu $\mathcal{R} = \|\Delta g\|^2 = \|\partial_{xx}g\|^2 + \|\partial_{yy}g\|^2$ minimalizuje křivost. Křivost je pro jednorozměrnou funkci rovna

$$\kappa = \frac{f''(x)}{(1 + f'(x)^2)^{3/2}}$$

a tím, že se minimalizuje druhá derivace $\partial_{xx}, \partial_{yy}$ se minimalizuje i velikost křivosti. Navíc minimum druhé derivace odpovídá minimální difúzi [16]. Velikost difúze je

$$\int \{\nabla \cdot D \cdot \nabla g(x, y)\}^2 dx dy$$

kde D je difúzní tenzor. Výsledek je, že nejhladší řešení je zároveň řešení s nejmenší difúzí.

Je možné použít i vyšší řády lineární regularizace, ale musí se počítat s tím, že konečné difference vyšších řádů už průměrují derivaci přes větší část matice a ztrácí význam lokální derivace.

2.1.2 Minimum Fisherovy informace - MFI

Vylepšením lineární regularizace prvního řádu je metoda hledání minima *Fisherovy informace* I_F . Matice Fisherovy informace je definovaná jako [17]

$$(J_X)_{i,j} = \left\langle \frac{\partial \ln f_X(x)}{\partial x_i} \frac{\partial \ln f_X(x)}{\partial x_j} \right\rangle = \int_{R^n} \frac{\partial \ln f_X(x)}{\partial x_i} \frac{\partial \ln f_X(x)}{\partial x_j} f_X(x) d^n x.$$

v dvourozměrném případě pro funkci $g(x, y)$ je možné získat velikost Fisherovy informace I_F

$$\begin{aligned} I_F &= \int \sum_{i,j=1}^2 \left| \frac{\partial g(x, y)}{\partial x_i} \frac{\partial g(x, y)}{\partial x_j} \frac{1}{g(x, y)} \right| dx dy \\ &= \sum_{i=1}^N \partial g_i \frac{1}{g_i} \partial g_i = \partial g^T W \partial g \quad W_{ij} = \delta_{ij}/g_i \end{aligned} \tag{2.8}$$

kde ∂g_i jsou součty velikostí konečných diferencí ve směru $\partial g_i = |\partial g_{ix}| + |\partial g_{iy}|$ a W je tzv. váhová matice.

Zajímavou vlastností I_F je Cramer-Raova nerovnost

$$\sigma^2 \geq \frac{1}{I_F}$$

ze které je vidět, že minimalizací Fisherovy informace roste dolní hranice možné střední kvadratické odchylky σ^2 rekonstruovaných dat. Zde je spojitost s entropií,

protože pokud mají data Gaussovo rozdělení, tak se zvětšující se σ^2 se zvětšuje entropie. Navíc lze říct [3], že řešení s minimem I_F je nejhladší možné řešení. Také je ukázáno, že tato metoda je kvalitnější, než je metoda hledání maxima entropie. Metoda je navíc dobře odolná vůči šumu.

Základní princip I_F je celkem jednoduchý, funguje stejně jako lineární regularizace jenom díky „vážení“ pomocí $1/g_i$ jsou umožněné vyšší hodnoty derivace ve vyšších hodnotách g_i a naopak v nízkých hodnotách na okraji plazmatu, které jsou méně přesné, jsou povolené jenom malé derivace a funkce je tedy hladší. Nevýhoda toho přístupu oproti ostatním lineárním regularizacím je v tom, že hodnota maxima může být nadhodnocena (nebo podhodnocena) oproti realitě, zatímco o ostatních lineárních regularizacích je jistota, že reálná hodnota bude vyšší.

2.1.3 Vážená druhá derivace - MVDR

Přístup metody MFI, kdy se derivace váží různým koeficientem lze zobecnit, podrobnější popis je v [2]. Váhová matice může mít tvar

$$W_{jj} = \begin{cases} (\bar{g}/g_j)^\beta & \text{pro } g_j \geq \gamma\bar{g} \\ 1 & \text{pro } g_j < \gamma\bar{g} \end{cases}$$

kde $\bar{g}$ je průměr velikosti g , β a γ jsou konstanty, jejichž typické hodnoty jsou $\beta = 1 - 3$ a $\gamma = 1 - 1,5$. Tato vylepšená verze má za cíl méně zkreslovat píky a používá se spolu s minimem velikosti druhé derivace. Nevýhoda oproti MFI je v tom, že neexistuje tak propracovaná matematická teorie a koeficienty se musí odhadnout z praxe. Jako další nevýhoda se ukázala občas horší konvergence metody než MFI.

2.1.4 Princip maximální entropie

Další používaný princip pro regularizaci matice je princip maximální entropie. Je to celkem logický přístup, pokud o určitém jevu není známo téměř nic, tak alespoň lze předpokládat, že bude nabývat maximální entropie. Výsledek by mělo být nejprůirozenější řešení.

Pro zjednodušení se počítá, že každý pixel g_j může nabývat diskrétních hodnot vzdálených ε . Výkon vyzářený jedním pixelem je roven

$$g_j = n_j \varepsilon$$

Navíc z celkového vyzářeného výkonu lze určit sumu n_i

$$\frac{I_0}{\varepsilon} = \sum_{j=1}^L n_j \equiv N$$

Potom množství přerovnání kvant ε lze vyjádřit jako

$$W = \frac{N!}{\prod_{j=1}^L n_j!}$$

dále pro $\varepsilon \ll 1$, tzn. $N \rightarrow \infty$, po dosazení do definice entropie mikrokanonického souboru

$$S = k \ln W = -k \sum n_j \ln n_j + C = -\frac{k}{\varepsilon} \sum g_j \ln g_j + C'$$

protože v tomto případě se hledá maximum, tak lze konstanty před sumou a za sumou zanedbat. Takto vypočtenou funkci jde použít jako hodnotící funkci do vztahu (2.6). Také je možné přidat další okrajovou podmínku, použitou ve výpočtu, a to je celkový vyzářený výkon

$$I_0 = \sum_{j=1}^L g_j$$

Nevýhoda tohoto algoritmu je vyšší výpočetní náročnost. Další nevýhoda je, že po hledané funkce není vyžadována lokální hladkost a výsledná funkce může být zašuměná náhodnými fluktuacemi, aby celkově funkce odpovídala maximální entropii.

2.1.5 Neuronové sítě

Trochu mimo stojí ještě metoda neuronových sítí. Neuronové sítě se skládají z jednoduchých procesorových jednotek (neuronů), které jsou mezi sebou propojeny určitým způsobem. Obvykle jsou propojeny do vrstev, kdy každý neuron z jedné vrstvy má spojení se všemi z vrstvy následující. Každý neuron má určitou přenosovou funkci, která pomocí vstupních dat určí výstup. Obecný tvar rovnice přenosu je tento

$$Y = S \left(\sum_{i=1}^N (w_i x_i) + \Theta \right)$$

kde $S(x)$ je přenosová funkce, w_i jsou váhové konstanty, x_i jsou vstupní data a Θ je práh interakce. Váhové konstanty w_i a práh interakce Θ je nutné určit postupným učením neuronové sítě a tento proces může být poměrně náročný. Výhodou neuronových sítí, pokud se je podaří úspěšně aplikovat na daný problém, vysoká rychlost rekonstrukce. Naopak nevýhoda je neschopnost rekonstruovat funkci, pokud je její tvar výrazněji odlišný od funkcí, na kterých byla neuronová síť vytrénována.

2.2 Porovnání algoritmů

Na závěr této kapitoly zde bude uvedeno stručné shrnutí výhod a nevýhod jednotlivých algoritmů. V rámci této práce byly testovány pouze pixelové metody lineárními regularizacemi a váženými lineárními regularizacemi. Vlastnosti ostatních algoritmů jsou popsány pouze na základě jiných zdrojů. Výsledné porovnání je v tabulce (2.1).

Analytické metody	+ nižší výpočetní náročnost – předpoklad určité kruhové/eliptické symetrie – potřeba přesné znalosti středu – nepočítá se s konečným počtem známých hodnot – špatná implementace a-priory znalostí
Radonova transformace	– nestabilní vůči šumu – v praxi se nepoužívá
Abelova transformace	+ pro rekonstrukci stačí pouze jeden detektor + velmi jednoduchá metoda + možnost analytického vyjádření inverzní funkce – předpoklad dokonale osově symetrického plazmatu
Cormackova metoda	+ plazma nemusí být přesně kruhové + odolnost vůči šumu – aliasing
Fourier-Besselova metoda	+ vylepšení Cormackovy metody
Pixelové metody	+ počítá se s konečným počtem známých hodnot měření + odolnost proti šumu + snadná implementace a-priory znalostí + snadné započtení geometrie – vyšší výpočetní náročnost
Lineární regularizace	+ rekonstrukce je vyhlazená + snadná úprava pro shlazování podle mag. pole – velikost píků je vždy podceněná
Nelineární regularizace (MFI, MVDR)	+ rekonstrukce je vyhlazená + velikost píků je přesnější + snadná úprava pro shlazování podle mag. pole – nutnost řešit iteračně, protože jsou nelineární
Abelizace (4.1.1)	+ pro rekonstrukci stačí pouze jeden detektor + lze použít pro libovolný tvar siločar + nízká výpočetní náročnost + možnost využití nelineární regularizace – předpoklad symetrie rekonstrukce podle siločar
Maximum entropie	+ odpovídá minimální entropii – rekonstrukce je zašuměná – vysoká výpočetní náročnost
Neuronové sítě	+ velmi rychlá metoda – dlouho trvající učení – umí pouze interpolovat naučené tvary

Tabulka 2.1: Porovnání různých algoritmů pro tomografii plazmatu

Kapitola 3

Numerické řešení Tichonovovy regularizace s minimem Fisherovy informace

V této kapitole je popsán numerický princip algoritmu, který se používá v ÚFP AV ČR a tím je Tichonovova regularizace s minimem Fisherovy informace. Tichonovova regularizace je obecně hledání minima funkcionálu ve tvaru

$$\|Ax - b\|^2 + \|\Gamma x\|^2 \quad (3.1)$$

kde Γ je Tichonovova matice a $Ax = b$ je nedostatečně určená soustava jejíž řešení se hledá. Je zřejmé, že hledání minima tohoto funkcionálu je ekvivalentní hledání extrému Lagrangeovy funkce (2.7), protože po dosazení za χ^2 a odstranění konstant, které nejsou při hledání extrému podstatné, funkce (2.7) přejde do tvaru

$$\Lambda = \|\tilde{T}g - \tilde{f}\|^2 + \lambda \|\Gamma g\|^2 \quad (3.2)$$

kde $\tilde{T}_{ij} = T_{ij}/\sigma_i$ a $\tilde{f}_i = f_i/\sigma_i$. Za matici Γg poté lze dosadit libovolnou lineární regularizaci z předchozí kapitoly. Místo vážení pomocí σ_i lze pro vážení využít inverzní kovarianční matici K [2, 16] a předchozí vztah zapsat jako

$$\Lambda = (Tg - f)^T K (Tg - f) + \lambda (\Gamma g)^T (\Gamma g)$$

Ale tento způsob není v aktuální verzi použitého algoritmu implementován. Nyní se počítá s tím, že chyby jednotlivých senzorů jsou nezávislé. Nalézt extrém rovnice (3.2) je celkem jednoduché. Nejdříve se rovnice převede do maticového zápisu

$$\begin{aligned} \Lambda &= (\tilde{T}g - \tilde{f})^T (\tilde{T}g - \tilde{f}) + \lambda (\Gamma g)^T (\Gamma g) \\ &= (g^T \tilde{T}^T - \tilde{f}^T) (\tilde{T}g - \tilde{f}) + \lambda (g^T \Gamma^T) (\Gamma g) \\ &= \tilde{f}^T \tilde{f} - \tilde{f}^T \tilde{T}g - g^T \tilde{T}^T \tilde{f} + g^T \tilde{T}^T \tilde{T}g + \lambda g^T \Gamma^T \Gamma g \end{aligned}$$

nyní je možné sečíst oba lineární členy, protože jejich výsledkem je číslo a platí $(\tilde{f}^T \tilde{T} g)^T = g^T \tilde{T}^T \tilde{f}$.

$$\Lambda = \tilde{f}^T \tilde{f} - 2g^T \tilde{T}^T \tilde{f} + g^T \tilde{T}^T \tilde{T} g + \lambda g^T \Gamma^T \Gamma g$$

pro nalezení extrému poté již stačí najít vektor g , pro které platí $\nabla_{\tilde{g}} \Lambda = 0$

$$\nabla_{\tilde{g}} \Lambda = -2\tilde{T}^T \tilde{f} + 2\tilde{T}^T \tilde{T} g + 2\lambda \Gamma^T \Gamma g = 0$$

z čehož se získá soustava lineárních rovnic pro vektor g

$$(\tilde{T}^T \tilde{T} + \lambda \Gamma^T \Gamma)g = \tilde{T}^T \tilde{f} \quad (3.3)$$

Nyní je řešení zapsané ve tvaru $Ax = b$, pokud je matice A invertibilní, tak lze řešení zapsat i jako $x = A^{-1}b$, ale tímto způsobem se numericky nevypočítává. Nalezená matice je pro $\lambda \geq 0$ pozitivně semidefinitní a pokud je $\lambda > 0$ a matice Λ je regulární, tak je matice soustavy (3.3) dokonce pozitivně definitní, což umožňuje použít k výpočtu některé pokročilejší algoritmy. Nyní se pro řešení podle dokumentace MatLab [18] používá Choleskyho faktorizace pro řídké matice. Podrobněji je řešení soustavy popsáno v sekci 5.8.

Aby platilo toto odvození, tak matice Γ může být pouze lineární. Problém je, že matice Fisherovy informace (2.8) lineární není.

$$I_F = \partial g^T W \partial g \quad W_{ij} = \delta_{ij} / g_i$$

To se dá obejít tím, že se bude hledat řešení iterativně. Nejdříve se najde řešení pro lineární regularizaci prvního řádu, což je vlastně I_F bez váhové matice W . V dalším kroku se poté za Γ dosadí celá matice I_F . Důkaz konvergence vychází spíše z praxe, protože vektor g konverguje k řešení velmi rychle, ke zkonvergování zpravidla stačí 3 cykly [19]. Na druhou stranu se často stává, že v prvních třech krocích algoritmus konverguje a ve čtvrtém se kořen ztratí a výsledná rekonstrukce je silně poškozená, což bylo ověřeno na reálných datech (kapitola 6), proto není vhodné používat více než 3-4 cykly.

Váhovou matici lze ještě vylepšit normováním na maximum vektoru g . Potom má matice W v n -tém kroku tvar

$$W_{ij}^{(1)} = \delta_{ij} \quad W_{ij}^{(n)} = \frac{\delta_{ij} \max_k [g_k^{(n-1)}]}{g_j^{(n-1)}} \quad (3.4)$$

Normování má za cíl hlavně zlepšit konvergenci, protože geometrická matice T_{ij} i vstupní data f jsou normována, aby velikosti T, f a g byly pro různé měření podobné. Na závěr výpočtu se samozřejmě výsledek odnormuje. Pokud se tedy shrne princip nalezení řešení, dojde se k tomu, že je potřeba použít dva vnořené cykly:

- Vnitřní cyklus bude dosazovat řešení rovnice (3.3), do okrajové podmínky $\chi^2 - 1 = 0$ a hledat Langrangeův multiplikátor λ , tak aby podmínka platila. Je možné přidat další okrajové podmínky. Potom by se hledalo N multiplikátorů, aby bylo splněno N okrajových podmínek, což lze snadno převést na hledání řešení $\mathbb{R}^N \rightarrow \mathbb{R}$ zapsáním rovnic $\sum_{i=1}^N (R_i)^2 = 0$, kde R_i jsou okrajové podmínky ve tvaru $f(x) = 0$. Ovšem důsledkem by bylo zpomalení konvergence a prodloužení rekonstrukční doby.
- Vnější cyklus poté mění váhovou matici, jak již bylo naznačeno ve vztahu (3.4).

Nezápornost a nulovost rekonstrukce

Další problém, který zde nebyl zmíněn, je započtení nezápornosti prvků g_i uvnitř tokamaku a nulovosti mimo. Odstranění záporných prvků je důležité, protože se počítá s tím, že plazma je pro měřené vlnové délky dokonale průhledné. Záporné hodnoty tedy nemají žádný fyzikální smysl. Nejrychlejší a nejjednodušší způsob je v jednotlivých krocích rekonstrukce nahradit záporné hodnoty nulami, tímto se zabrání, aby měly záporné hodnoty vliv na velikost hodnoty χ^2 . Potom, ale rekonstrukce v okolí těchto opravených hodnot nemusí být úplně správná (hladká).

Další možnost by byla přidat další okrajovou podmínku

$$\sum_{g_i < 0} g_i = 0$$

Poté by bylo možné použít způsob výpočtu pro více okrajových podmínek zmíněný výše. Nevýhoda tohoto přístupu je nutnost poměrně rozsáhlé úpravy algoritmu oproti výpočtu pouze jedné podmínky, protože je nutné použít Newtonův algoritmus pro $\mathbb{R}^N$, což by pravděpodobně vedlo ke zhoršení konvergence a prodloužení výpočtu.

V článku [16] je zmíněna další možnost, kdy se upravují zdrojová data f_i tak, aby žádné nulové hodnoty nevznikaly. Základním principem je přidání dalšího členu l do rovnice (3.3), který mění levou stranu rovnice

$$(\tilde{T}^T \tilde{T} + \lambda \Gamma^T \Gamma)g = l + \tilde{T}^T \tilde{f}$$

V prvním kroku se položí $l = 0$, v dalších krocích se vypočítává korekce Δl jako

$$(\tilde{T}^T \tilde{T} + \lambda \Gamma^T \Gamma)\Delta g = \Delta l$$

kde $\Delta g_j = -g_j$ pro $g_j < 0$ a pro ostatní je nulové, případně se vyberou jenom body lokálních minim. Výsledkem je řešení bez záporných hodnot, ale upravují se upravují zdrojová data, což se může negativně projevit ve zbytku rekonstrukce. Výhodou může být to, že se nemění rekonstrukční matice a na výpočet Δg se nemusí znovu řešit celá soustava (Choleskyho faktorizace), ale stačí použít pouze zpětný chod.

Poslední možnost, která byla použita v této práci, je metoda virtuálních senzorů, jejichž zorné pole se mění každý cyklus podle toho, které hodnoty jsou záporné a každý senzor „hlídá“ jeden záporný bod, aby v něm byla nula. Přesností (velikostí

σ^2) virtuálního senzoru lze volit sílu s jakou je nezáporné řešení prosazováno. Navíc jsou záporné hodnoty ještě v každém kroku nulovány, aby neovlivňovali rekonstrukci, přesněji hledání řešení $\chi^2 = 1$.

Velmi podobně se dá vyřešit i nulování dat mimo tokamak. Určí se hranice tokamaku a body mimo jsou zvoleny za zorné pole virtuálních senzorů, které je nulují. Pro urychlení výpočtu stačí volit nulové jenom body blízko hranice a mimo tokamak body vynulovat až na závěr rekonstrukce.

3.1 Časové rozlišení (Rapid verze)

Existuje jednoduchá možnost, jak velmi zrychlit výpočet vektoru g tím, že řešení vypočte pro větší množství po sobě následujících snímků. Nejdříve je potřeba vyjádřit rovnici (3.3) ve tvaru

$$g = Mf$$

Toho lze snadno dosáhnout, pokud se označí

$$U = \tilde{T}^T \tilde{T} + \lambda \Gamma^T \Gamma \quad B = \tilde{T}^T \tilde{E}$$

kde $\tilde{E}_{ij} = \delta_{ij}/\sigma_i$, potom $M = U^{-1}B^1$. Nyní se ale nehledá matice M pro jednu rekonstrukci, ale pro všechny snímky. Proto je potřeba upravit podmínku (2.2)

$$\chi^2 = \frac{1}{S} \frac{1}{L} \sum_{\tau=1}^S \sum_{i=1}^L \left(\frac{f_{i\tau} - \sum_{j=1}^N T_{ij} g_{j\tau}}{\sigma_i} \right)^2$$

kde S je počet časových snímků a L počet známých projekcí f_i . Projekce f_i jsou navíc normované svým maximem v jednotlivých časech, aby při výpočtu neměly větší váhu časy s vyšší svítivostí, na konci výpočtu jsou samozřejmě vypočtené hodnoty $g_{j\tau}$ zpětně vynásobeny maximem f_i . Potom se hledá λ tak, aby byla splněna podmínka $\chi^2 = 1$. Do váhové matice W se dosadí časový průměr vektoru g , a dále se postupuje stejně jako u jednoho snímku. Jediná další závislost na naměřených datech v matici U je právě číslo λ a matice B nezávisí na datech. Takto určená matice M má výhodu, že se v ní zprůměruje šum jednotlivých měření a hlavně, že je pro všechna měření stejná a pro získání rekonstrukce stačí vynásobit $Mf = g$. Otázka přesnosti takovéto rekonstrukce je diskutována v [19] a v kapitole 6.

Lze se alespoň pokusit odhadnout, kolikrát bude tato metoda rychlejší za předpokladu, že počet senzorů L je mnohem menší než počet pixelů, kde počet pixelů je pro rozlišení $N \times N$ roven N^2 . Potom pro vypočtení S snímků samostatně je potřeba $O(SN^6)$ operací. Pokud se to bude počítat pro S matic dohromady, tak se prodlouží doba výpočtu χ^2 , ale to je složitost $O(N)$, takže nás to nezajímá. Matice M se počítá jako pro jeden snímek, takže složitost je $O(N^6)$ a na závěr se vypočtení řešení $g_\tau = Mf_\tau$ pro všechny snímky $O(SN^4)$, takže také zanedbatelné. Celkově to znamená, že výpočet by měl být přibližně S -krát rychlejší, ale samozřejmě je to

¹Zde se neřeší přímo inverzní matice, ale hledá se řešení soustavy rovnic s více pravými stranami $UM = B$

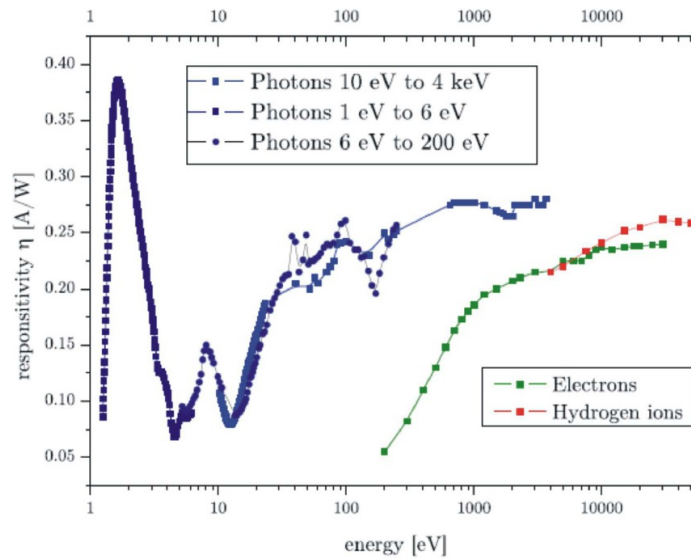
pouze řádový odhad. V praxi to vychází přibližně $S/3$ -krát rychlejší. Nevýhoda této metody je v tom, že MatLab umí využít při výpočtu matice M pouze jedno jádro procesoru, proto se nedá paralelizací urychlit výpočet jedné hromadné rekonstrukce.

Kapitola 4

Geometrie

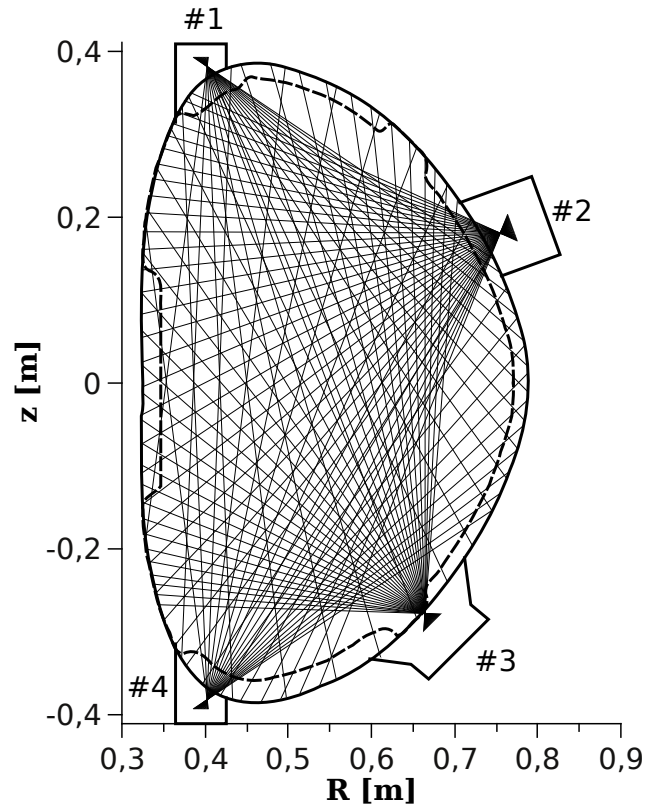
Tomografie plazmatu bude v tokamaku COMPASS využívat dva zdroje dat. Prvním zdrojem, který bude používán, je bolometrie. Účel bolometrie je obecně měřit vyzářený výkon plazmatu a díky tomografii lze získat navíc i prostorové rozlišení hustoty vyzářeného výkonu. Jako senzory pro bolometrii se budou používat AXUV (Absolute XUV) sondy, to jsou křemíkové fotodiody, které zachytávají spektrum záření od 0,0124 nm do 1100 nm (1 eV až 10 MeV) a lze je použít i k detekci nízkoenergetických elektronů a iontů. Velkou výhodou použití AXUV sond je nízká reakční doba, která je přibližně 0,7 ns [20], díky tomu lze získat vysoké časové rozlišení.

Naopak nevýhoda AXUV sond, stejně jako všech křemíkových detektorů, je nestejnoměrná citlivost na různé vlnové délky, což je viditelné právě v oblasti bolometrie (obr. 4.1), což je komplikace při výpočtu reálně vyzářeného výkonu. Většina energie přijímaná bolometrií není způsobena brzdným zářením, ale vyzařováním nečistot převážně v UV spektru a na začátek UV spektra (~ 10 eV) mají sondy nižší citlivost.



Obrázek 4.1: Typická citlivost diod η na záření, elektrony a ionty, zdroj [21]

Druhým zdrojem dat pro tomografii bude SXR diagnostika, což je diagnostika měkkého rentgenového záření. Pro SXR diagnostiku se používají také AXUV sondy,



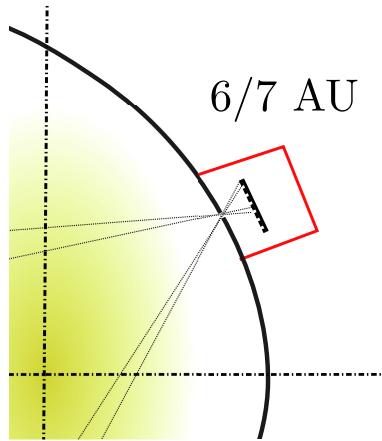
Obrázek 4.2: Schéma diagnostiky tokamaku COMPASS, zobrazeny chordy pro SXR diagnostiku

- #1 VUI- vertical upper inner
- #2 AU - angular upper
- #3 AL - angular lower
- #4 VLI - vertical lower inner

jenom se před ně vloží $5,1\,\mu\text{m}$ tlustý beryliový plíšek, který filtruje záření o nižší energii než $0,5\,\text{keV}$. Další rozdíl oproti bolometrii je v tom, že fotodiody pro SXR jsou zapojené v opačném směru než pro bolometrii [21].

Pro účely bolometrie a SXR diagnostiky byla v tokamaku COMPASS vymezena sekce 6/7 kde budou obsazeny čtyři porty. Z důvodu malých diagnostických portů tokamaku COMPASS bylo potřeba vyrobit úchyty senzorů přesně na míru. Porty AU a AL (obr. 4.2) budou obsahovat dvě AXUV diody po 20 kanálech. Dále jeden SXR detektor s 35 kanály a nakonec VIS diagnostiku (visible plasma radiation measurement), která je určena pro měření nečistot a okrajů plazmatu během výboje. VIS diagnostika bude mít 37 kanálů a je řešená přes optická vlákna vyvedená ven z detektoru. Porty VUI a VLI budou obsahovat jeden detektor buď AXUV nebo SXR, ale mezi jednotlivými experimenty bude možné detektory vyměňovat. V době psaní této práce byl osazen pouze port v poloze 6/7 AU (angular upper viz. 4.2). Ale v budoucnu budou osazeny všechny čtyři diagnostické porty.

Detektory jsou v portu umístěné tak, aby pokryly celý poloidální řez plazmatem. Ale protože cílem je získat co nejlepší prostorové rozlišení, tak musí každý kanál



Obrázek 4.3: Camera Obscura nebo také pinhole detector, schéma je pouze ilustrativní

detektoru vidět jinou část plazmatu, proto se před detektory umístí úzká štěrbina, která pomůže zostřit obraz na detektor, tzv. Camera Obscura (obr. 4.3). Šířka štěrbiny pro COMPASS je 0,1 mm a délka je 2 mm. V poloidálním směru šířka 0,1 mm zabezpečuje dostatečně prostorové rozlišení a ostrost, naopak v toroidálním směru byla zvolena délka 2 mm jinak by intenzita měkkého rentgenového záření nemusela být dostatečná pro detektory.

Skrz štěrbinu je obraz promítaný na pole detektorů. Umístění štěrbiny bylo určeno tak, aby byly minimalizovány překrytí zorných polí jednotlivých chord [21]. Chordy jsou přímky, podél kterých detektor pozoruje plazma. Pozorovací úhel je poměrně malý, takže přestože mají chordy určitou rozbíhavost (ve středu plazmatu 1-3 cm), tak se s nimi často jako s přímkami počítá, záleží na kolimaci svazku a velikosti pixelů.

4.1 Geometrická matice

Jak již bylo zmíněno v druhé kapitole, pro určení emisivity g_j je potřeba vyřešit soustavu

$$f_i = \sum_{j=1}^N T_{ij} g_j$$

kde f_i jsou naměřená data, g_j je hledaná emisivita a T_{ij} je geometrická matice, která zahrnuje vliv j -tého pixelu na i -tý detektor. A právě určení matice T_{ij} je cílem této kapitoly.

Za předpokladu, že je výkon v plazmatu vyzařován izotropně a za předpokladu, že je plazma pro měřené vlnové délky dokonale průhledné, pak je výkon detekovaný jedním detektorem roven

$$P_i = \int_{S_j} \int_{\mathbb{R}^+} \frac{\Omega_i(r)}{4\pi} G(r, \nu) n(\nu) dV d\nu$$

což je součet vyzářeného výkonu $G(r, \nu)$ uvnitř kužele S_j . Navíc je zapotřebí započíst různou citlivost detektoru pro různé frekvence pomocí funkce $n(\nu)$ a úhel, pod kterým vidí plazma detektor $\Omega_i(r)$. Díky zachování propustnosti (étendue) tedy součinu prostorového úhlu Ω a plochy A , která je kolmá na jeho osu, lze předchozí integrál upravit

$$P_i = \frac{(A\Omega)_i}{4\pi} \int_{l_i} \int_{\mathbb{R}^+} G(s, \nu) n(\nu) ds d\nu$$

tedy od objemového integrálu přejít k dráhovému integrálu po přímce l_i . Dále, protože detektor nemá stejnou citlivost pro různé vlnové délky, tak se místo $n(\nu)$ se započte střední hodnota $\langle n(\nu) \rangle$ ¹. Pokud se navíc zadefinuje vztah

$$f_i = \frac{4\pi P_i}{(A\Omega)_i} \quad (4.1)$$

jako světlost chody i , tak lze přejít ke vztahu pro světlost

$$f_i = \langle n(\nu) \rangle \int_{l_i} g(r) dr$$

kde $g(r) = \int_{\mathbb{R}^+} G(r, \nu) d\nu$. Dále lze nahradit integraci po přímce funkcí, která je nenulová v místě přímky l_i . Tady je nutné dodat, že pokud by se nepočítalo s rozbíhavostí chordy, tak by bylo potřeba použít Diracovu delta funkci (tedy distribuci) jako ve vzorci (1.2).

Pokud se počítá s rozbíhavostí, tak lze ve třech rozměrech nahradit přímku kuželem nebo spíše ve dvou rozměrech kruhovou výsečí, která má už konečné hodnoty po celé ploše. Musí se ale započítat pokles vlivu $g(r)$ se zvětšující se vzdáleností od štěrbin jako $1/r$, což plyne buď ze zachování propustnosti $(A\Omega)$ nebo také z představy výseče jako množství přímek, které se rozbíhají a jednotlivých přímek je vliv konstantní. Takováto funkce, která zahrnuje vliv na i -tý detektor, bude označena $T_i(x, y)$. Díky tomu přejde předchozí integrál do tvaru

$$f_i = \int T_i g dx dy \quad (4.2)$$

Dále je potřeba ověřit, jestli je opravdu funkce $T_i(x, y)$ konstantní v určité vzdálenosti nebo jestli po krajích klesá. Pokud by byla štěrbina bodová, tak bude pokrytí rovnoměrné. Pokud štěrbina není bodová, tak pokrytí nebude rovnoměrné, jak je vidět na obrázku (4.4). Uprostřed bude pokrytí rovnoměrné a po stranách bude lineárně klesat. Jednoduchým geometrickým výpočtem lze spočíst, že poměr plochy, kde není intenzita konstantní vůči celé ploše je roven

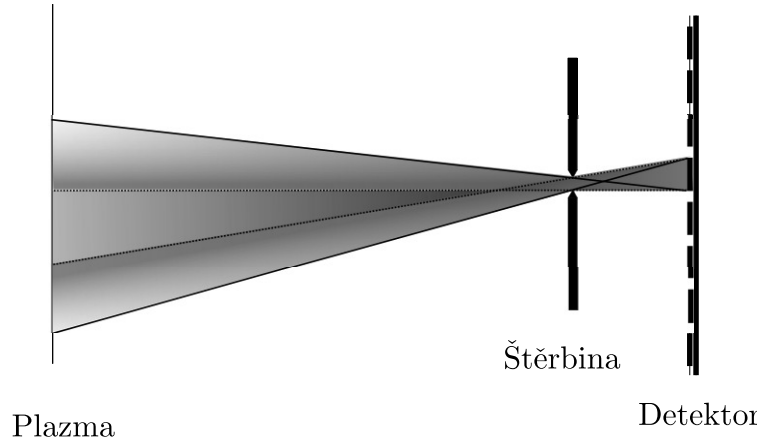
$$\eta = \frac{2\Delta}{\Delta + hL/(L + D)}$$

¹Střední hodnota přes rozdělení $G(s, \nu)$

kde Δ je šířka štěrbin, D je vzdálenost detektoru od štěrbin, h je šířka jedné fotodiody a L je vzdálenost od detektoru do středu plazmatu. Pro tokamak COMPASS a jeho detektor AL AXUV1 jsou tyto hodnoty v tabulce (4.1). Poměr η vychází 26%, tedy 13% na každé straně.

	[mm]
Δ	0,1
h	0,8
L	288
D	10

Tabulka 4.1: Parametry detektoru AL AXUV1



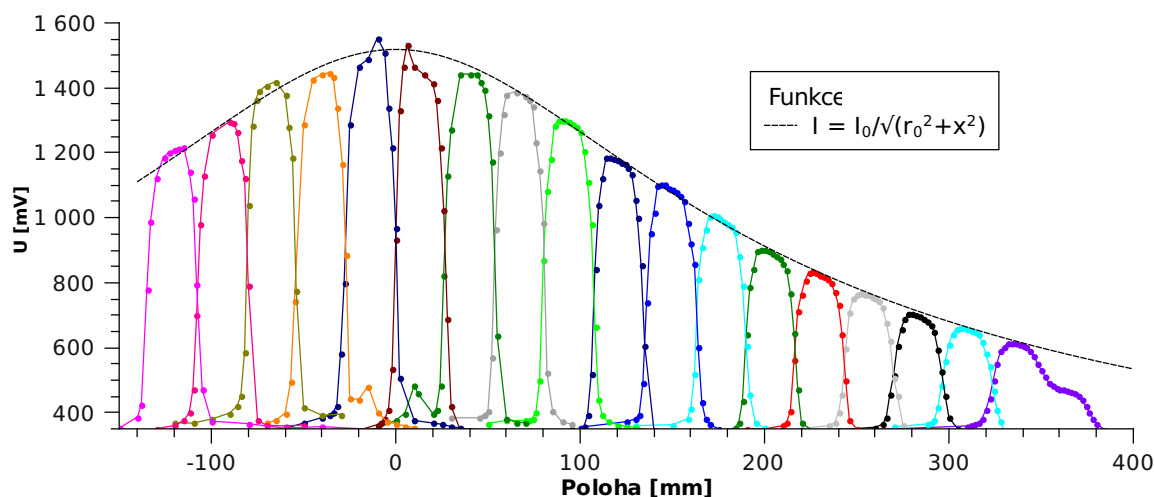
Obrázek 4.4: Pokrytí plazmatu z jednoho kanálu (bez ohybových jevů), poměry jsou pouze ilustrativní

Navíc u bolometrie mohou nastávat ohybové jevy. Z Rayleighova kritéria plyne, že rozbíhavost svazku po průchodu štěrbinou je

$$\alpha \approx \frac{\lambda}{\Delta}$$

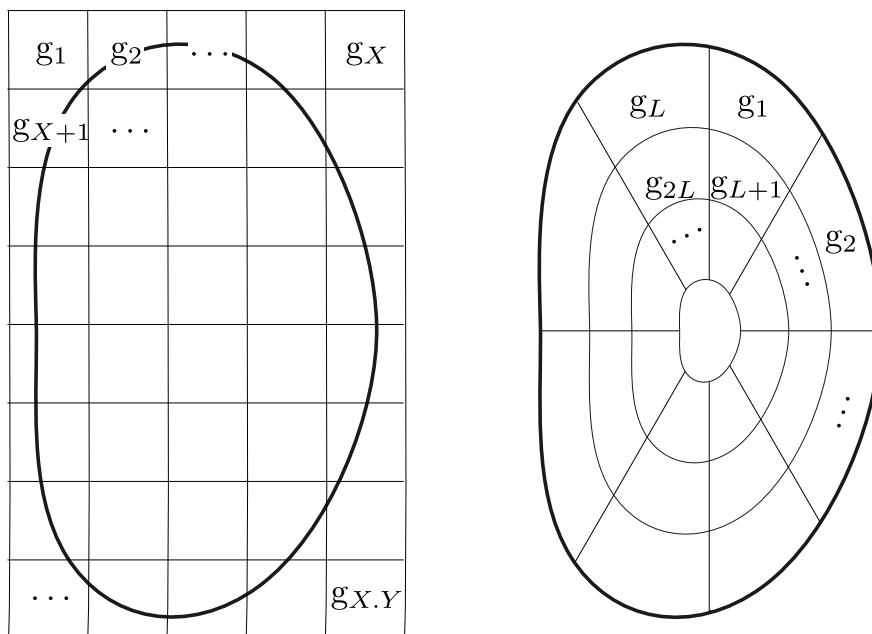
kde λ je vlnové délka záření a Δ je šířka štěrbin. Pokud se do této podmínky dosadí vlnová délka UV záření 300 nm a šířka štěrbin, potom úhel $\alpha = 3 \cdot 10^{-3}$ rad. Pro porovnání úhel pod jakým je ze štěrbin vidět jedna fotodiody je $13 \cdot 10^{-3}$ rad, pokud je fotodiody přímo naproti štěrbině. Pokud je fotodiody mimo osu, tak je průmět šířky štěrbin menší, takže α se zvětší, a naopak úhel pohledu je menší, takže poměr vyjde ještě hůř. Tyto údaje ovšem platí pouze pro bolometrii, u měkkého rentgenového záření ohyb nemá žádný vliv. Důsledkem toho nelze rozpoznat jevy v plazmatu, které jsou menší než třetina šířky jedné chordy, samozřejmě té nejužší, která vidí daný bod.

Vliv nebodové štěrbin je v algoritmu započítán, přestože pro rozlišení 50×50 px je maximální šířka chordy ~ 4 px a lineární pokles pro COMPASS nastává pro 10% plochy na každé straně, proto jsou korekce na nehomogenitu chordy pro běžná rozlišení zbytečné. Avšak pro jiné tokamaky už mohou být korekce podstatnější.



Obrázek 4.5: Kalibrační graf senzoru AXUV2, kde různé fotodiody jsou značeny odlišnou barvou. Poloha byla měřena v laboratoři a udává vzdálenost na přímce, kde byl umístěn bodový zdroj světla pro kalibraci [21], v poloze 0 mm byl zdroj kolmo na detektor přímo naproti štěrbíně.

V dalším kroku je potřeba rozložit plochu, na které se vypočítává funkce g , na diskrétní pixely. Pixely je možno volit libovolně, aby pokryly celou plochu a zároveň se nepřekrývaly. Lze volit například tvary pixelů jako soustředné kružnice nebo vzít pixely ve směru magnetického pole, ale takovouto volbou se udělá velmi silný předpoklad na tvar plazmatu. Pokud se pixely zvolí čtvercové nebo obdélníkové, všechny stejné velikosti, tak se nedává žádný výrazný předpoklad pro tvar plazmatu. Příklady možných uspořádání jsou na obrázku (4.6).



Obrázek 4.6: Příklady možných uspořádání pixelů v tokamaku

Další důvod, proč volit jednoduché obdélníkové pixely, je z programátorského hle-

diska, protože je mnohem jednodušší například přepsat matici obdélníkových pixelů do vektoru (matice $n \cdot n \times 1$) nebo vytvořit matici derivace. To jsou důvody, proč se v algoritmu pro tomografii COMPASS předpokládá s použitím obdélníkových pixelů. Plocha poloidálního řezu tokamakem se rozdělí na pravidelnou síť $nx \times ny$ pixelů a zářivost plazmatu v jednotlivých pixelech se označí jako g_{ij} . Pro potřeby výpočtu se síť g_{ij} přepíše do vektoru g_k ($k \in \{1, \dots, nx \cdot ny\}$) například se vektor přerovná, aby $k = nx \cdot j + i$. Potom lze funkci (4.2) přepsat do tvaru

$$f_i = \sum_{j=1}^{nx \cdot ny} \int_{S_j} T_i(x, y) g(x, y) dx dy$$

kde S_j je plocha pixelu j . Za předpokladu, že jsou pixely dost malé, lze tvrdit, že vliv bodů jednoho pixelu na určitý detektor je po celé ploše pixelu konstantní, tento vliv se označí T_{ij} . To nemusí platit pro pixely velmi blízko detektoru, ale tam by se stejně plazma nemělo vyskytovat. Za $g(x, y)$ se potom dosadí střední hustota vyzářeného výkonu na ploše pixelu, tím se dostane konečný vztah

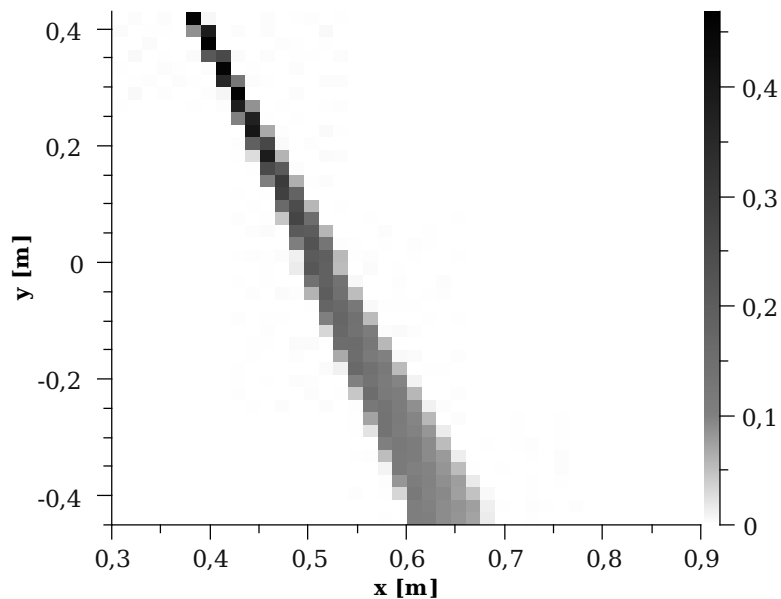
$$f_i = \sum_{j=1}^{nx \cdot ny} T_{ij} g_j$$

Nyní je potřeba určit geometrickou matici T_{ij} . Pokud by se počítala precizně, tak bylo nutné spočítat plochu každého pixelu, která je vidět z jednotlivých detektorů a určit průměrnou délku pixelu, kterou vidí detektor a započítat i to, že je nemusí vidět celá fotodioda (obr. 4.4).

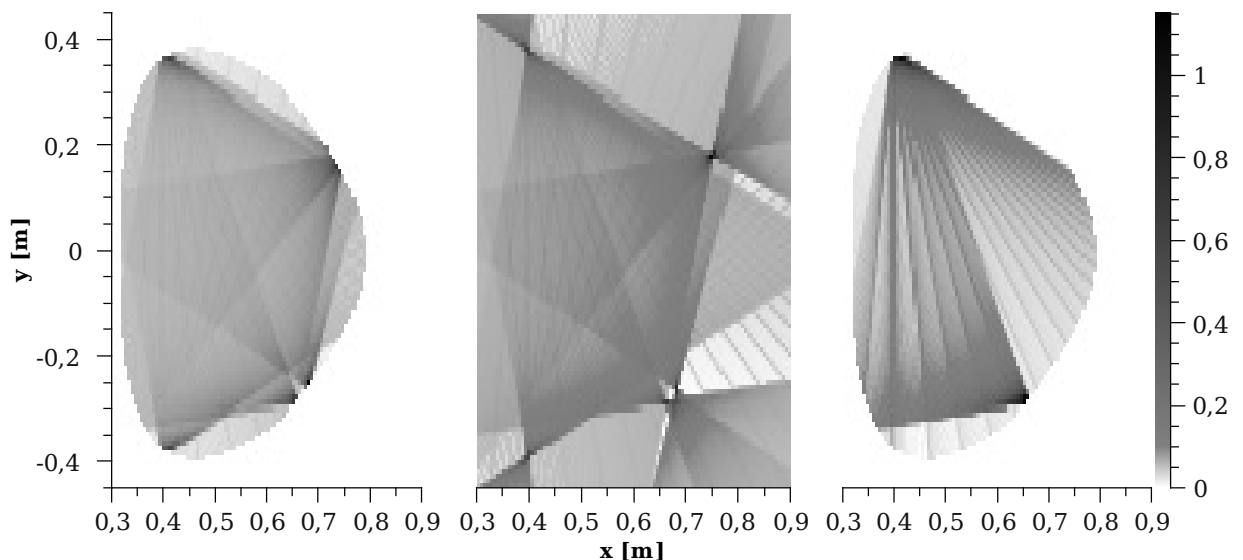
Velmi jednoduchá možnost výpočtu je vytvořit přímky, která vycházejí z jednotlivých detektorů (chordy) a zvolit velikosti T_{ij} jako délku chordy z detektoru i , která prochází pixelem j . Pokud byly hodnoty světlosti chordy f_i absolutně kalibrovány (rov. 4.1), tak potom hodnoty g_i odpovídají plošné hustotě vyzářeného výkonu.

Další věc, kterou je nutné do matice T_{ij} zahrnout, je rozbíhavost zorného pole. Pokud je dobrá kolimace nebo malé rozlišení rekonstrukce, tak je to možné zanedbat. U bolometrie a měření SXR na COMPASSu je rozbíhavost vyšší a proto je vhodné ji započíst. Započtení rozbíhavosti lze jednoduše udělat pomocí představy, že z jednotlivých detektorů vychází více virtuálních chord, které rovnoměrně pokrývají celé zorné pole. Pokud se zvolí počet virtuálních chord dostatečně vysoký, aby byl každý pixel zahrnutý nejlépe vícekrát, tak bude pokrytí zorného pole chordy rovnoměrné a velikost jednotlivých pixelů bude úměrná ploše. Výsledná chorda je v grafu (4.7).

Velké množství virtuálních chord nevlivní dobu rekonstrukce g_i , protože všechny virtuální chordy jednoho detektoru se sloučí do jednoho řádku matice T_{ij} . Další výhodou je, že změnou hustoty virtuálních chord lze zahrnout jevy jako nenulovou šířku šterbiny. Naopak nevýhodou oproti přístupu, kdy se použije jediná přímka, je v tom, že matice T_{ij} bude mít mnohem více nenulových prvků a to může zpomalit výpočet, podrobněji je tento problém diskutován v sekci 5.5. Na první pohled se sice zdá, že použít jedinou chordu je moc velké zanedbání, ale protože algoritmus hledání minima Fisherovy informace obsahuje vyhlazování, tak se rekonstrukce nemusí až tak moc lišit. Obecně platí, že pro SXR by se měl použít úplný profil chordy [2].



Obrázek 4.7: Detail jedné sloučené chordy detektoru VUI vypočtené pomocí metody virtuálních chord



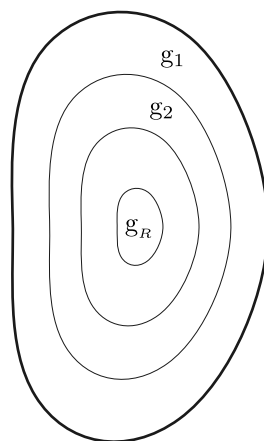
Obrázek 4.8: Výsledná matice T_{ij} po sečtení příspěvků od všech detektorů a transformování zpět do matice. Na prvním grafu je celá matice používaná pro rekonstrukci, na druhém je matice bez ořezání a na posledním je pouze detail detektorů ze dvou portů

V algoritmu přiloženému k této práci (příloha [A](#)) na výpočet geometrické matice T_{ij} se pro výpočet používají virtuální chordy a hodnoty prvků matice jsou rovny průměrné délce protnutí chordou. Pro rychlý výpočet geometrické matice se používá upravený Xiaolin Wu algoritmus [\[22\]](#) s tím, že výpočet antialiasingu je pozměněn, aby velikosti hodnot v prvcích matice odpovídaly přesně protnuté délce. Na závěr se virtuální chordy sloučí a matice se vydělí počtem virtuálních chord.

Další problém, který bylo nutné v bakalářské práci vyřešit, jsou body mimo tokamak. To je v použitém algoritmu (příloha A) řešené celkem jednoduše. Projdou se všechny pixely a pro každý pixel se projdou všechny body na hranici. Nejdříve se spočítá vektor mezi aktuálním pixelem a prvním bodem hranice, ten je označený vecA . Dále se určí vektor vecB mezi aktuálním pixelem a dalším bodem na hranici a udělá se jejich vektorový součin vecC . Podle jeho výsledku se určí, jestli vektor leží napravo nebo nalevo od vecA a porovná se s vektorem nejvíce nalevo a napravo. Po průchodu všech bodů na hranici se porovnají vektory nejvíce vlevo a vpravo, a pokud je mezi nimi úhel menší než 180° , potom tento pixel leží mimo tokamak. V tom případě se vynuluje bod j pro všechny detektory. Matici před a po ořezání lze vidět na grafu (4.8).

4.1.1 Kruhový profil pixelů, tzv. „Abelizace“

Trochu mimo stojí ještě 1D profil (obr. 4.9). V tomto případě se počítá se symetrií emisivity plazmatu v poloidálním směru v jednotlivých magnetických plochách.



Obrázek 4.9: Matice pro „abelizaci“, pixely opisují tvar magnetického pole

Pro tuto geometrickou matici platí všechny věci zmíněné v předchozí sekci jako započítání rozbíhavosti chord. Výpočet této matice je jednoduchý. Stačí určit délky průsečíků jednotlivých chord se všemi „mezikružícími“, což obecně mohou být libovolné vnořené konvexní tvary. Největší výhodou této „abelizace“ je v tom, že k rekonstrukci stačí pouze jeden senzor. Navíc oproti abelizaci zmíněné v sekci 1.2 má výhodu v tom, že lze využít fungující algoritmus pro minimalizaci Fisherovy informace pouze s drobnými úpravami. Algoritmus tvorby geometrické matice a rekonstrukce je přiložený k této práci (příloha A). Na druhou stranu se podařilo rekonstruovat emisivitu z jednoho senzoru i pomocí matice derivace ve směru magnetického pole, což je výhodné v tom, že jsou na tvar plazmatu kladeny mnohem slabší požadavky.

4.2 Matice derivace

4.2.1 Derivace ve směru

Jak již bylo několikrát zmíněno pro rekonstrukci emisivity g je potřeba využít dalších informací jako minimum Fisherovy informace nebo minimum první nebo druhé derivace. Z toho důvodu je potřeba vytvořit matici derivace. Protože nejsou známy hodnoty funkce g ve všech bodech, ale jen v jednotlivých pixelech, potom je nutné nahradit derivaci konečnými diferencemi. Podrobně je toto téma popsáno v [16], ale zde je uveden jednodušší rozbor. Jako derivaci lze použít buď zpětnou diferenci (popř. dopřednou diferenci)

$$f'_i = \frac{f_i - f_{i-1}}{\Delta x}$$

nebo centrální diferenci

$$f'_i = \frac{f_{i+1} - f_{i-1}}{2\Delta x}$$

Výhodou centrální difference je nižší chyba derivace, která u centrální difference klesá s druhou mocninou délky intervalu (šířky pixelu) oproti tomu zpětná difference klesá jenom s první mocninou. Na druhou stranu má zpětná difference lokálnější charakter a to je zvláště pro nižší rozlišení podstatné. Nejjednodušší tvar matice derivace je tedy tento

$$B = \begin{pmatrix} -1 & 1 & 0 & 0 & \cdots \\ 0 & -1 & 1 & 0 & \cdots \\ & & \ddots & & \\ \cdots & 0 & 0 & -1 & 1 \end{pmatrix}$$

Toto je ovšem tvar pouze pro derivaci ve směru x . Pro derivaci ve směru y nebo případně pro šikmé derivace se musí kladné prvky posunout. Pokud je počet pixelů g ve směru x roven X potom bude mít zpětná difference ve směru y tvar

$$f'_i = \frac{f_i - f_{i-X}}{\Delta x}$$

Popřípadě pro šikmé směry se odečítá člen $f_{i-X\pm 1}$. Druhou derivaci lze potom vyjádřit jako

$$f''_i = \frac{f_{i-1} - 2f_i + f_{i+1}}{\Delta x^2}$$

Druhou derivaci lze snadno vyjádřit jako $-B^T B$, kde B je první derivace, je to tedy záporně vzatá velikost první derivace.

$$B = \begin{pmatrix} -2 & 1 & 0 & 0 & \dots \\ 1 & -2 & 1 & 0 & \dots \\ & & \ddots & & \\ \dots & 0 & 0 & 1 & -2 \end{pmatrix}$$

Další problém jsou prvky na okraji pixelového pole g . Na kraji se odčítá poslední pixel v j -té řadě od prvního v $j + 1$ řadě. To se dá řešit vynulováním těchto problémových diferencí v matici derivace, což je korektní řešení. Nevýhoda tohoto řešení je, že matice (3.3) už nemusí být pozitivně definitní, ale bude pouze pozitivně semidefinitní a nebude existovat jednoznačné řešení. S tím souvisí i to, že MatLab po detekci singulární matice použije místo Choleskyho faktorizace jiný algoritmus výpočtu, který je mnohonásobně pomalejší [18].

Druhá možnost je ignorovat problémy tohoto tvaru matice derivace, protože okolo hranice tokamaku v matici pixelů je z každé strany alespoň jeden pixel, který je nulovaný. Důsledek je, že derivace okrajových pixelů je nulová a není potřeba okrajové derivace řešit.

4.2.2 Derivace podle magnetického pole

Při tomografické rekonstrukci lze využít znalost tvaru magnetického pole uvnitř tokamaku. Ve směru magnetických siločar (matice $B_{\parallel}$) se emisivita mění méně, než ve směru kolmém (matice $B_{\perp}$) na siločáry. Pokud se tedy vytvoří tyto matice derivace, tak lze vzájemným vážením určit, jak moc se která matice podílí na hladkosti. Takové vyhlazení se nazývá anizotropní, zatímco klasické vyhlazení ve směru os x, y se nazývá izotropní. Většinou se poměr $B_{\perp}/B_{\parallel}$ volí 1-0,01 [2]. Nevýhoda takovéto matice je menší flexibilita oproti obdélníkové síti, protože se do rekonstrukce vnáší další předpoklad, který nemusí být přesný nebo pravdivý. Na druhou stranu výsledná rekonstrukce může být mnohem přesnější. V algoritmu, který je přiložený k této bakalářské práci (příloha A), se konstruuje matice první derivace, ale není problém algoritmus upravit na generování druhé derivace, aby výsledek odpovídal minimální difúzi (2.1.1).

Postup výpočtu matice derivace podle siločar je následující. Nejdříve se ve středu každého pixelu interpoluje směrnice magnetického pole. V dalším kroku se směrnice rozloží do směrových derivací ve směrech $\{k\pi/4 \mid k \in \{0, \dots, 7\}\}$. Směrnice je vždy rozložena do jedné složky šikmé derivace (a) a jedné složky (b) rovnoběžné s jednou s hlavních os podle vzorců

$$a = \sin(2\alpha)/\sqrt{2} \quad b = \cos(2\alpha)$$

kde α je relativní úhel v rámci daného oktantu a je počítaný od osy oktantu, která je rovnoběžná s osou x nebo y . Normování pomocí $\sqrt{2}$ je kvůli šikmé derivaci, zde se pro jednoduchost počítá, že pixely jsou přibližně čtvercové. Matice kolmé derivace k magnetickému poli se vytvoří otočením směrů derivace o 90° . Na závěr se vždy každý řádek normuje, aby suma mimodiagonálních prvků byla 1 a na diagonále

bylo číslo -1. Výsledná matice může rekonstruovat mnohem přesněji bez preference kosočtvercových tvarů, kterou lze pozorovat při aplikaci obyčejné matice derivace. Navíc byla ověřena možnost použití pouze jednoho senzoru pro rekonstrukci dat.

Naopak jedna z nevýhod této metody plyne z podrobnějšího prozkoumání matice derivace. Problém je v tom, že jednotlivá „mezikruží“ jsou spojena poměrně silně, ale mezi vnitřními kruhy je už spojení slabé a může se stát, že při špatně zvolené matici derivace se algoritmu vyplatí vytvořit nerealistický dutý profil (artefakt), aby vyhověl podmínce na MFI.

Kapitola 5

Numerické simulace

Další důležitá oblast tomografie plazmatu jsou numerické simulace. Numerickou simulací se myslí vygenerování umělého průřezu emisivity plazmatu, který je následně rozložen podle geometrické matice, a jsou určeny hodnoty, které vidí jednotlivé detektory.

$$\sum T_{ij}g_j = f_i$$

Toto je vhodná metoda pro studování různých algoritmů, vyhodnocování chyb, stability při šumu nebo například systematických chyb v geometrii.

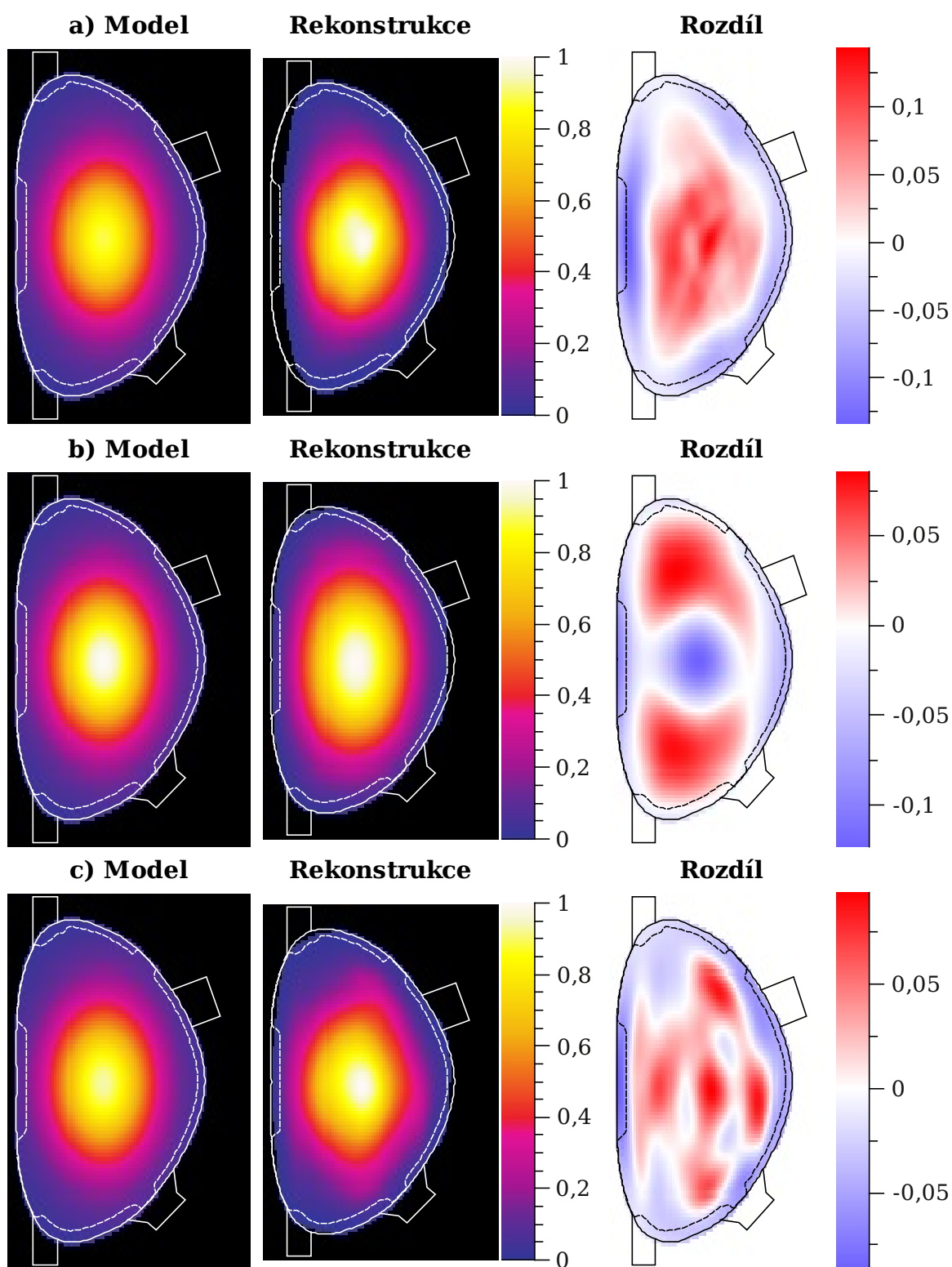
5.1 Porovnání algoritmů

První věc, kterou je důležité ověřit, jsou chyby vlastních algoritmů. V této sekci budou testovány regularizační matice pro minimum Fisherovy informace (MFI), minimum druhé derivace (MDR) a nakonec minimum vážené druhé derivace MVDR zmíněné v sekci 2.1.3.

První testovaný průřez je Gaussova křivka ve dvou rozměrech. Toto je celkem základní průřez a všechny algoritmy ho rekonstruují bez větších problémů, jak je vidět na (obr. 5.1). Emisivity byly normovány, aby suma jejich výkonu byla stejná. Na (obr. 5.1a) se zdá, že rekonstrukce MFI je nesymetrická směrem vyšší hustotě pokrytí senzory (obr. 4.4), ale v dalších rekonstrukcích bylo ověřeno, že asymetrii, která preferuje kosočtvercový tvar, způsobuje matice derivace ve směru os x, y . Pokud se použije derivace ve směru magnetického pole (obr. 5.9), tak se asymetrie částečně odstraní. Jinak je rekonstrukce (obr. 5.1a) bez problémů.

Rekonstrukce pomocí MDR je viditelně podceněná uprostřed plazmatu a naopak moc velká na okraji, na druhou stranu je rekonstrukce velmi symetrická. Poslední rekonstrukce (obr. 5.1c) pomocí vážené druhé derivace je poměrně symetrická a má nejmenší rozdíl od původní funkce.

Další průřez emisivity (obr. 5.2) jsou dvě Gaussovy křivky, které se od sebe odečítají. Vzniklý dutý průřez emisivity je poměrně těžké rekonstruovat a jsou na něm vidět různé vady použitých algoritmů. Na prvním grafu se MFI podařilo určit to, že je



Obrázek 5.1: Zobrazení rekonstrukcí Gaussovy křivky, a) MFI, b) MDR, c) MVDR

průřez dutý. Na dalším grafu se MDR ani prakticky nepodařilo rekonstruovat střed a výška rekonstrukce je podceněná. Na posledních grafech (obr. 5.2c) se vážené druhé derivaci MVDR podařilo rekonstruovat střed a zároveň je rekonstrukce méně asymetrická a také rozdíl od původní funkce je nejmenší.

5.2 Použití různé velikosti šumu

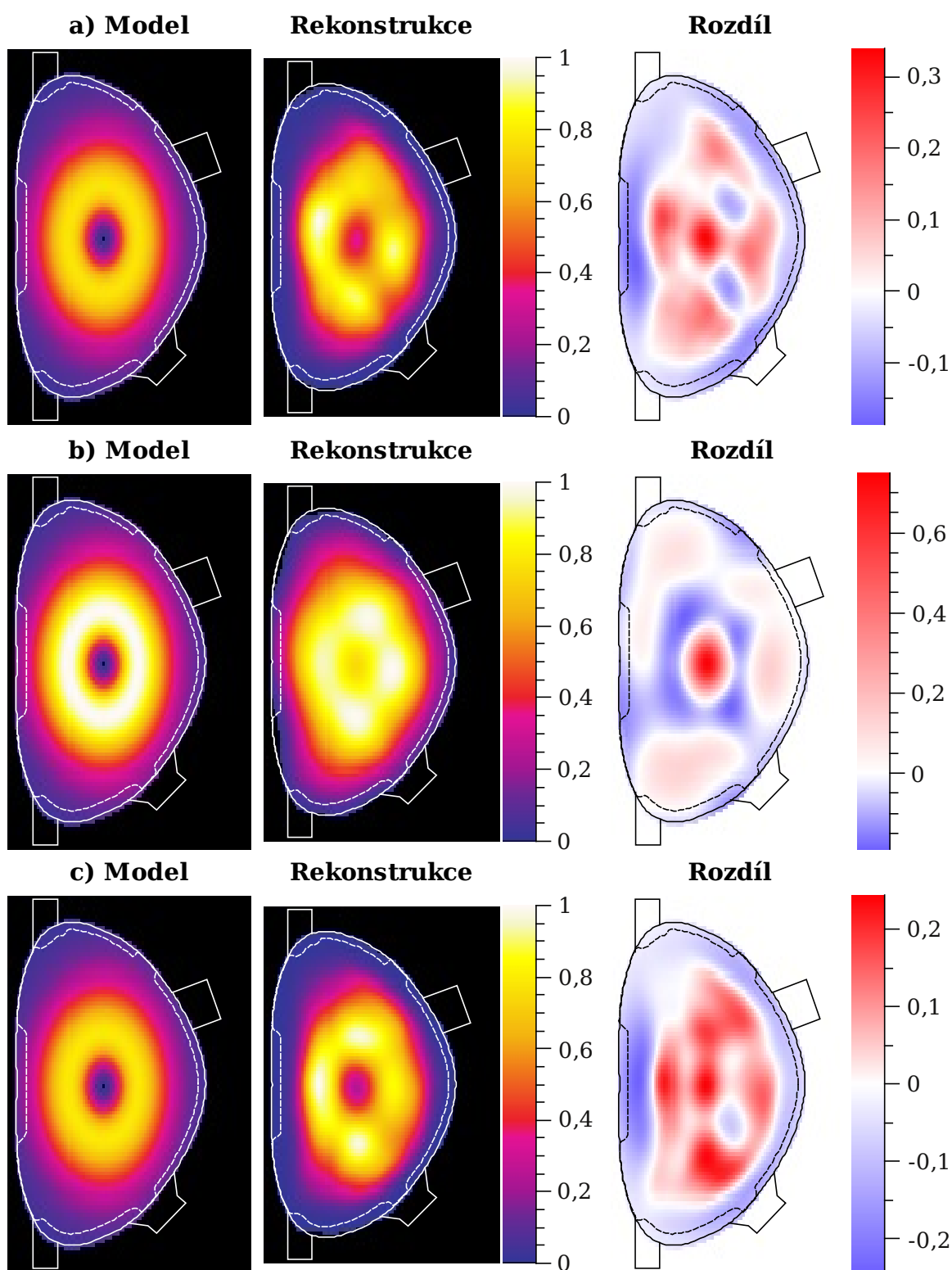
Další velmi důležitá vlastnost algoritmu je odolnost proti šumu. Šum v detektorech se předpokládá do 5%, spíše méně. Proto byl pro numerické simulace vytvořen v „naměřených“ datech Gaussovský šum a pro jednoduchost je simulovaný šum ve všech detektorech stejně velký. V rámci testování vlivu šumu na rekonstrukci byl použit algoritmus vážené druhé derivace. Algoritmus minima Fisherovy informace měl podobné chování pro různý šum, ale horší rekonstrukci modelové funkce pro nízký šum. Jako modelovou funkci byl použit dutý průřez emisivity testovaný v úvodu této kapitoly (obr. 5.2). U rekonstrukcí s vyšší hodnotou šumu ($\geq 5\%$) je nutné počítat s tím, že pro různé náhodné hodnoty šumu se může rekonstrukce poměrně výrazně lišit.

Na grafu (obr. 5.3) je vidět, jak se pro vyšší šum ztrácí detaily. Proto je zajímavé porovnat, jak bude rekonstrukce kvalitní pro různé velikosti vyhlazování pro šum dat 5%. Na dalším grafu (obr. 5.4) se rekonstrukce emisivity s 5% šumem pro 4% vyhlazení opravdu zhorší a naopak pro 6% vyhlazení se ztrácí detaily rekonstrukce a rekonstrukce vypadá jako s 10% šumem. Zde je vidět, jak je důležité přesně určit velikost šumu, jinak může dojít k výraznému zhoršení rekonstrukce.

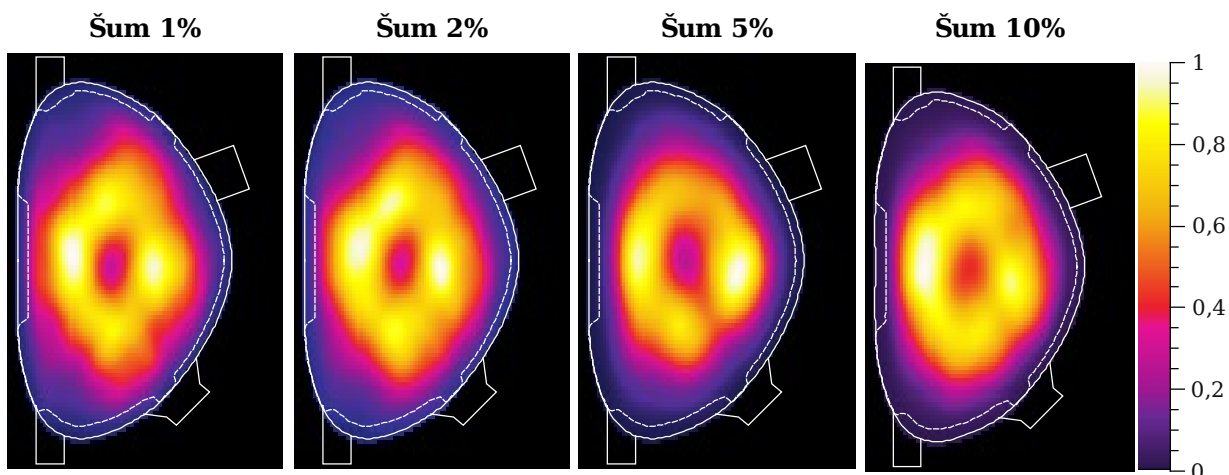
5.3 Systematická chyba

Polohy detektorů v tokamaku nejsou známy zcela přesně, a přestože se polohy senzorů kalibrují, tak stále nemusí přesně odpovídat realitě například kvůli teplotní roztažnosti tokamaku. Proto je důležité ověřit, jak moc poškodí posunutí detektoru rekonstruovanou emisivitu. V této simulaci má geometrická matice, podle které je funkce rekonstruovaná, posunutou šterbinu VLI detektoru o 0,5 mm, takže odchylka chord ve středu plazmatu je přibližně 3,5 cm směrem vpravo. Na grafu (obr. 5.5) je vidět, že se opravdu emisivita posunula doprava a zdeformovala, aby odpovídala posunutí chord a maximální deformace dosahovala až 15% oproti rekonstrukci bez poškození. Na druhou stranu posun menší než 0,1 mm se na rekonstrukci projevil minimálně, změna tvaru emisivity byla zanedbatelná a maximální chyba v okolí středu emisivity byla $\sim 2\%$.

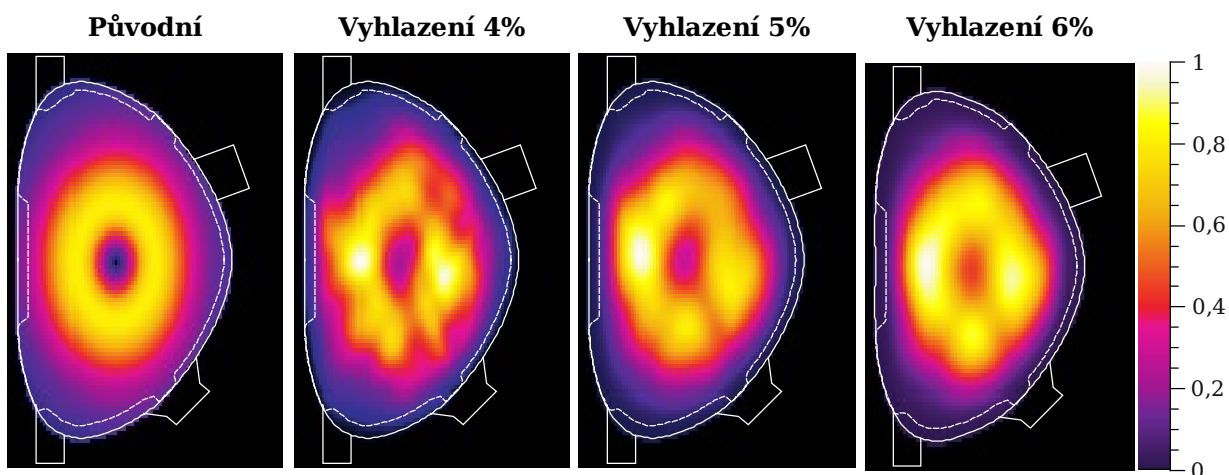
Množství systematických chyb, které lze testovat je velmi mnoho. Navíc některé systematické chyby se mění časem jako různá citlivost detektorů například kvůli radiačnímu poškození, jiné zase mohou vzniknout za určitých podmínek jako posun detektorů způsobený tepelným rozpínáním tokamaku, v tomto případě se buď počítá, že posun stejný a chyby se vystředují nebo je nutné provést kalibraci tak, aby byly posuny kompenzovány.



Obrázek 5.2: Zobrazení rekonstrukcí dutého průřezu, a) MFI, b) MDR, c) MVDR



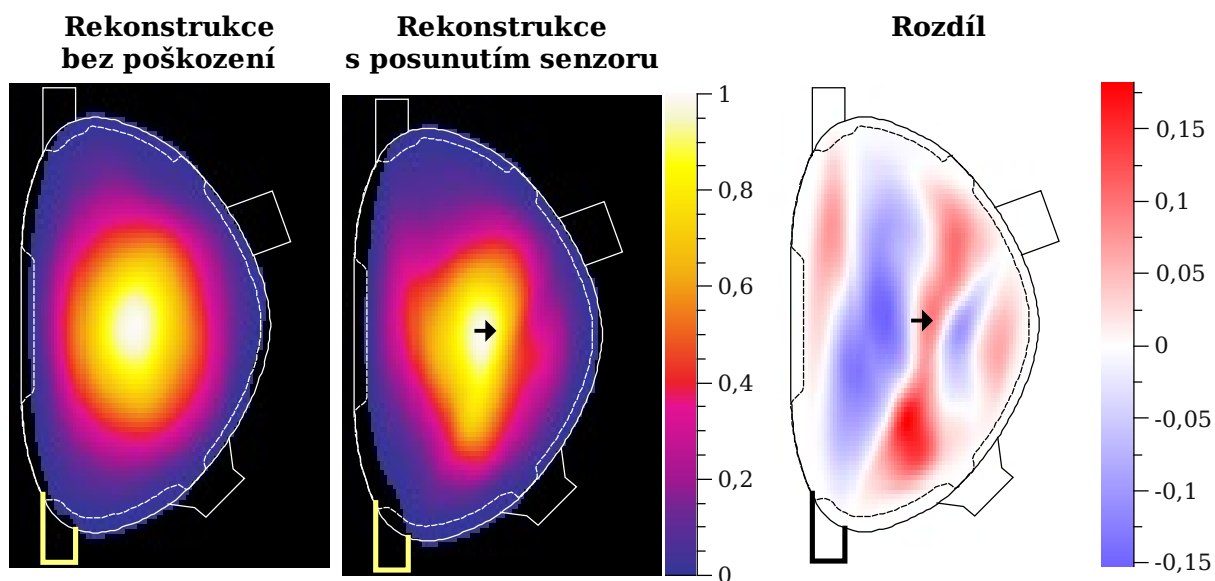
Obrázek 5.3: Rekonstrukce dutého průřezu emisivity pro různé úrovně šumu. Velikost šumu dat odpovídá velikosti σ^2 očekávaného detektoru.



Obrázek 5.4: Rekonstrukce dutého průřezu emisivity pro šum detektorů 5% s různou úrovní vyhlazování (předpokládaného šumu)

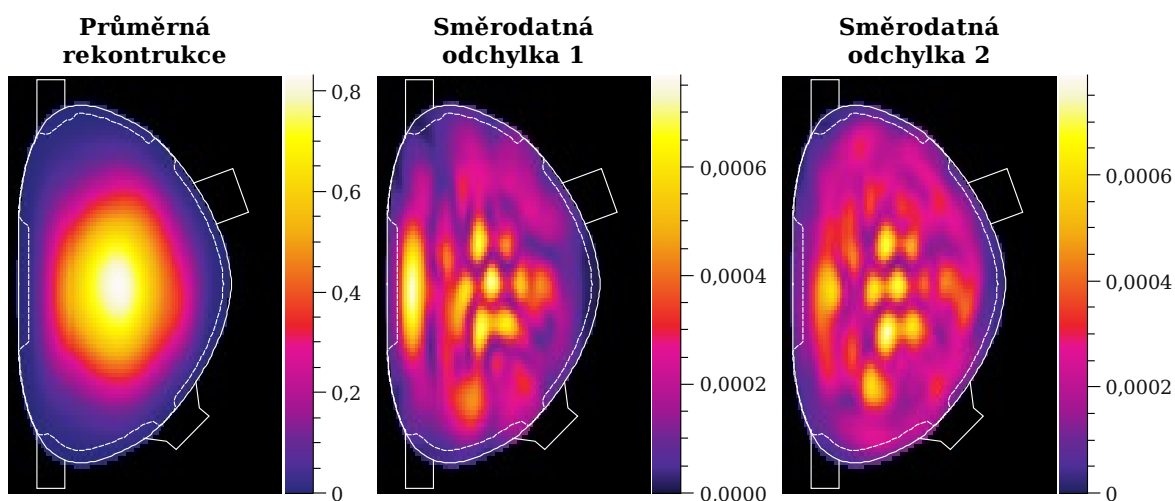
5.4 Počáteční podmínky

V definici váhové matice W (3.4) se předpokládalo, že v prvním kroku se místo $1/g_j$ dosadí 1. Pokud se ovšem použije nějaký předpoklad na tvar rekonstruované funkce, jako například Gaussova křivka nebo v časové rekonstrukci předchozí snímek, tak je možné dosadit už v prvním kroku tento odhad. Tento krok by měl zlepšit konvergenci a tedy i rychlost řešení. Zajímavý je ale případ, kdy se rekonstruovaná funkce velmi liší od předpokladu. Na grafu (obr. 5.6) je rekonstrukce Gaussovy křivky a jako předpokládaný původní tvar je sice Gaussova křivka, ale se středem umístěným v náhodném bodě v tokamaku. Na grafu je vidět, že rozptyl 10 rekonstrukcí je okolo 0,1% s tím, že maximum je vlevo od středu plazmatu, což je místo, kde vidí nejméně detektorů, takže šlo předpokládat, že tam se problémy projeví. Navíc se při testování rekonstrukce ukázalo, že použití různých předpokladů vede k výraznému



Obrázek 5.5: Zobrazení rekonstrukce při posunutí štěrby o 0,5 mm v senzoru VLI (zvýrazněný), vzniklé posunutí chord o 3,5 cm uprostřed plazmatu je naznačeno šipkou

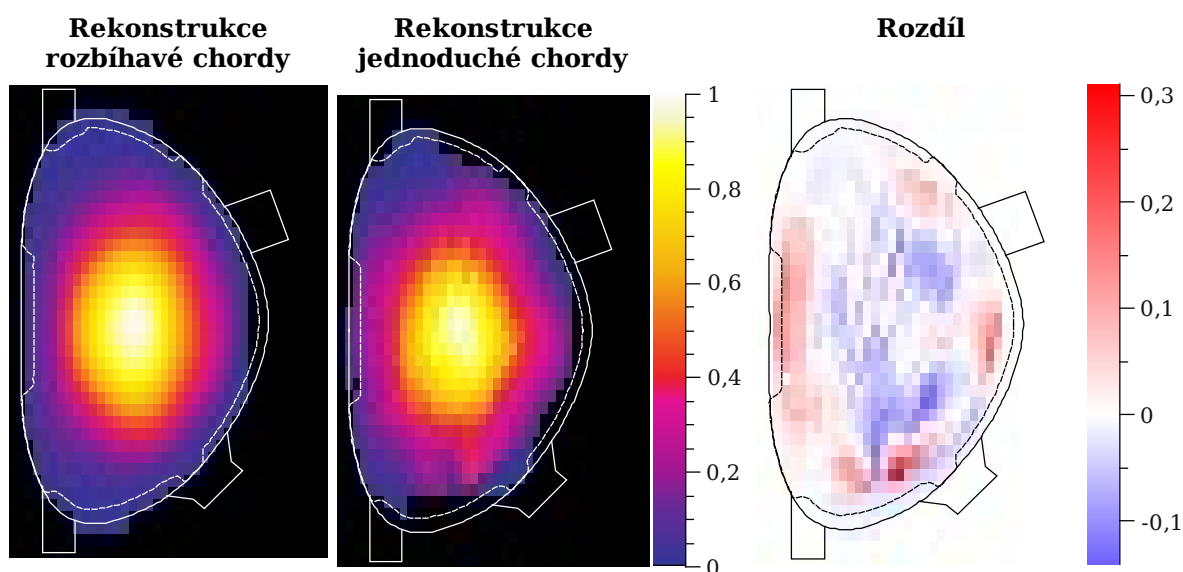
zlepšení, ale občas i zhoršení konvergence. Pokud se za předpoklad dosadí přímo správný výsledek, tak funkce nezkonverguje hned v prvním kroku, protože Gaussova křivka neodpovídá minimu Fisherovy informace. Také je zajímavé, že zvýšením počtu vnějších cyklů se odchylka téměř nezměnila, jediné zmizel výrazný pík v levé části rekonstrukce. Přestože konvergence prvních tří cyklů je poměrně rychlá (konvergence ve vnějším cyklu z praxe vychází lepší než $1/2^n$), naopak pro vyšší množství iterací zůstávají změny g mezi jednotlivými vnějšími iteracemi téměř stejné.



Obrázek 5.6: Rekonstrukce a směrodatná odchylka při 3 a 5 vnějších cyklech MFI

5.5 Použití jednoduchých chord

V kapitole 4 Geometrie bylo zmíněno, že v některých případech lze zanedbat rozbíhavost chord. Výhodou tohoto přístupu by měla být vyšší rychlost výpočtu, protože geometrická matice bude řidší. Vyšší rychlost řešení jednotlivé matice se opravdu projevila. Pro rozlišení 40×40 pixelů se snížila hustota geometrická matice z 10% na 4,5% a průměrná doba řešení matice klesla z 0,19s na 0,14s. Bohužel tato úprava velmi zhoršila konvergenci algoritmu, takže celková doba rekonstrukce se prodloužila o 10% a pro vyšší rozlišení algoritmus nezkonvergoval vůbec. Navíc, aby algoritmus zkonvergoval, tak bylo potřeba desetkrát zvýšit očekávané chyby rekonstrukce (vyhlazení) σ^2 . Výsledná rekonstrukce bez šumu je v (obr. 5.7) a lze vidět, že i kvalita rekonstrukce je horší.



Obrázek 5.7: Rekonstrukce s použitím rozšířených a jednoduchých chord

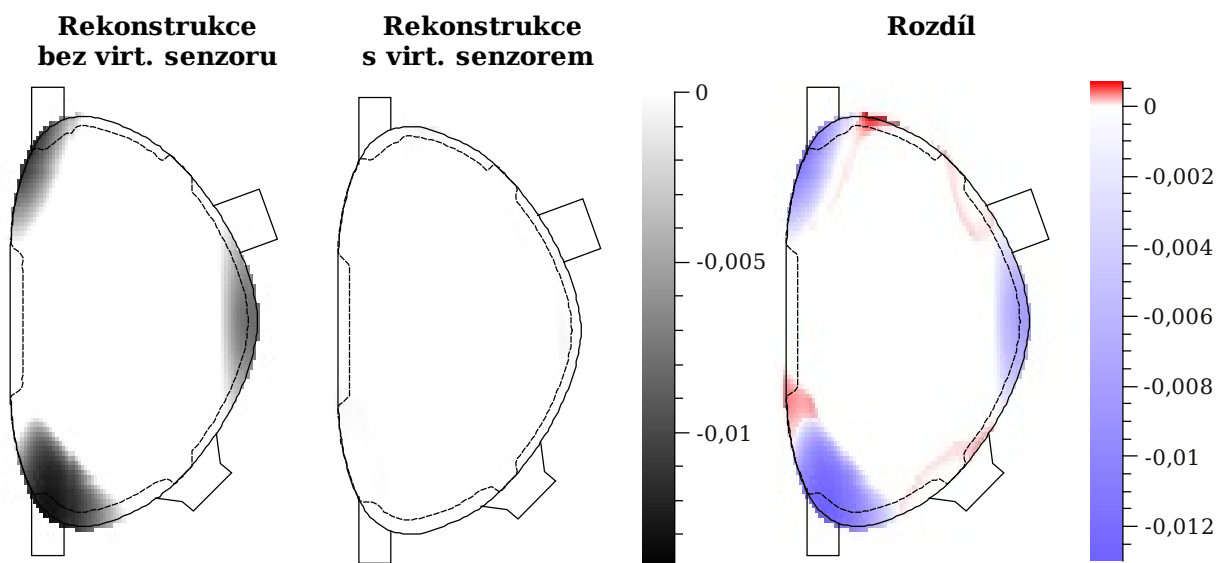
5.6 Podmínka nezápornosti vyzařování

Jedna z užitečných informací, které lze při rekonstrukci použít je nezápornost vyzařování. Zde popsáný algoritmus MFI v sobě nemá žádnou podmínku, která by preferovala nezáporné řešení, ale jednoduchou úpravou lze získat částečnou preferenci nezápornosti. Váhová matice má již zmíněný tvar

$$W_{ij}^{(1)} = \delta_{ij} \quad W_{ij}^{(n)} = \frac{\delta_{ij} \max_k [g_k^{(n-1)}]}{g_j^{(n-1)}}$$

Tím, že se za $g_j \leq 0$ zvolí malá kladná konstanta se zabrání dělení nulou a vedlejší efekt je silné vyhlazení záporné části. V některých případech ani toto nestačí, zvláště při použití vyhlazování pomocí druhé derivace a menších rozlišení matice. Poté je

nutné použít některou další z metod, které byly zmíněny v sekci 3. Pro tento případ byla použita metoda virtuálních senzorů. Každou vnější iteraci se určí záporné prvky a ty jsou zvoleny za zorné pole virtuálních detektorů, které je vidí nulové. Problém je v tom, že když se zvolí zorné pole virtuálních detektorů pro určité prvky, tak se v daném kroku vynulují, ale okolo těchto prvků se mohou objevit další záporné prvky. V dalším kroku je proto potřeba označit nové a zároveň potlačovat záporné z předchozího kroku. Následkem toho se zhorší konvergence ve vnějším cyklu a je lepší zvolit více kroků, alespoň 5. Výsledná rekonstrukce je na grafu (obr. 5.8), bylo nutné použít poměrně umělý průřez emisivity s velmi strmým poklesem. Záporné hodnoty se podařilo potlačit na méně než na 1% původní velikosti na většině plochy.



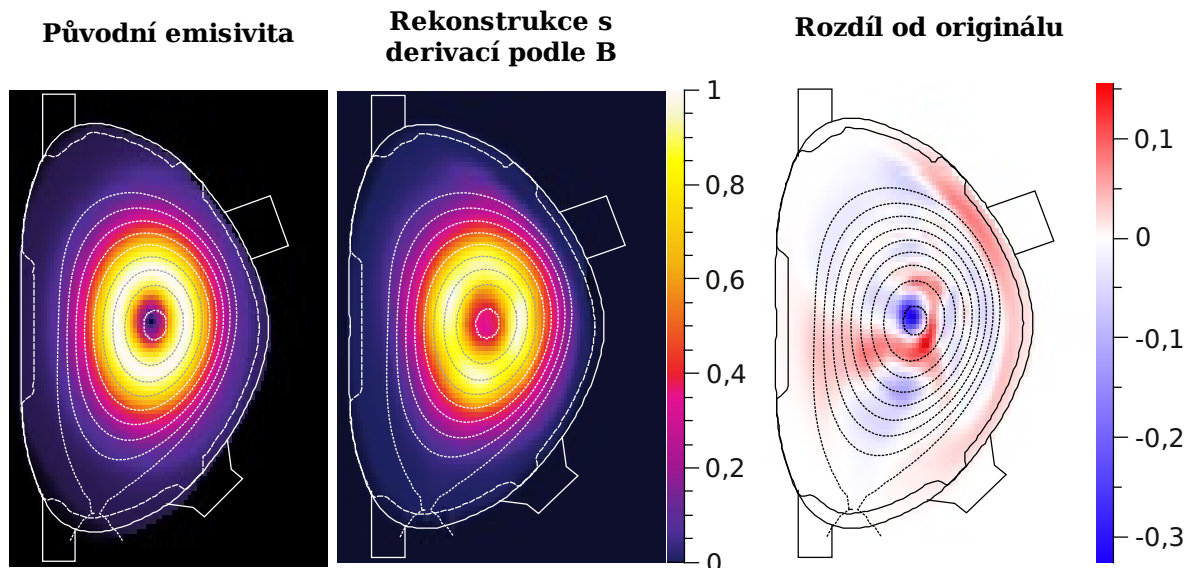
Obrázek 5.8: Záporné hodnoty rekonstrukce při použití virtuálního senzoru. Modrá barva rozdílu značí zlepšení.

5.7 Matice derivace podle magnetického pole

V sekci 4.2.2 je zmíněna možnost použití znalosti rozložení magnetického pole ke zlepšení rekonstrukce. Algoritmus pro generování matice derivace je přiložený k práci (příloha A). Správnou volbou vážení mezi maticí derivace rovnoběžné s magnetickým polem $B_{\parallel}$ a maticí kolmou na magnetické pole $B_{\perp}$ lze dosáhnout poměrně dobrých výsledků viz (obr. 5.9)

Nevýhodou je předpoklad znalosti tvaru magnetického pole, které může vnést do rekonstrukce další chyby, což je vidět na středu rekonstrukce (obr. 5.9), kde je střed původní emisivity záměrně mírně posunut vůči magnetickému poli. Rekonstrukce je už shlazená přesně podle pole. Výhodou je odstranění „kosočtvercového“ tvaru preferovaného obyčejnou derivací ve směru os a navíc je takto možné rekonstruovat emisivitu i z pouze jednoho senzoru.

Další nevýhoda je vyšší hustota bodů matice a tedy nižší rychlost řešení. Oproti jednoduché matici derivace se doba výpočtu průměrně zdvojnásobila.



Obrázek 5.9: Rekonstrukce dutého průřezu emisivity s 5% šumem při použití matice derivace podle magnetického pole $B_{\parallel}$ a kolmo na magnetické pole $B_{\perp}$

5.8 Optimalizace algoritmu

Při použití algoritmu hledání MFI je často potřeba vytvořit větší množství rekonstrukcí například pro různé statistické vyhodnocení numerických simulací. V těchto případech je důležité, aby byl algoritmus co nejrychlejší. Původní verze algoritmu používala jenom několik optimalizačních metod jako například hledání řešení pomocí metody sečen. Jediná možnost, jak provést rekonstrukci většího množství měření naráz bylo využít toho, že v rámci jednoho měření se data moc neměnila a použít způsob časové rekonstrukce popsany v sekci 3.1.

Jiná možnost je vytvořit rekonstrukci pro každý časový snímek zvlášť. Z toho důvodu bylo potřeba výrazně zrychlit výpočet. Původní algoritmus provedl v MatLabu rekonstrukci Gaussovy křivky v rozlišení 40×40 pixelů na dvoujádrovém procesoru 1,86 GHz za 70s, pokud by se udělalo 10000 rekonstrukcí, tak by výpočet trval 8 dnů.

Podle profileru MatLabu více než 95% doby výpočtu trvá řešení soustavy (3.3), proto existují pouze tři způsoby, jak rekonstrukci urychlit. Je možné urychlit jednotlivé řešení soustavy, snížit množství iterací, kdy se řeší tato soustava, nebo snížit rozlišení rekonstrukce, protože složitost výpočtu je $O(N^6)$, podrobněji v sekci 3.1.

5.8.1 Řídké matice

Prvním a velmi podstatným zlepšením bylo použití řídkých matic v celém algoritmu místo plných. Tímto krokem se ušetřilo velké množství paměti. Pro rozlišení 40×40 potřeboval původní algoritmus 250 MB, po upgradu už pouze jednotky MB a pro rozlišení 100×100 pixelů byla spotřeba paměti větší než 3 GB a po upgradu 400 MB.

Také řešení a hlavně násobení řídkých matic je mnohem rychlejší. Po použití řídkých matic v MatLabu se snížila doba rekonstrukce ze 70 s na 25 s.

5.8.2 Hledání kořene

Dalším urychlením je vylepšení metody sečen pro hledání řešení okrajové podmínky $\chi^2 = 1$. V rámci hledání nejlepší metody byly otestovány různé algoritmy jako například Regula-Falsi [23]. Výhoda Regula-Falsi je v tom, že nemůže minout řešení a přitom teoretická rychlost konvergence je podobná jako u metody sečen. Naopak výhoda metody sečen je v možnosti využití znalosti λ z předchozího kroku a tím urychlit řešení.

Velmi dobře se osvědčila kombinace půlení intervalu pro nalezení bodu $\chi^2 - 1 < 0$ a Regula-Falsi v prvním kroku, v dalších krocích potom lze použít metodu sečen. Důvod, proč je nutné řešit hledání kořene takto složitě, je v tvaru závislosti $\chi(\lambda)$. Funkce je sice spojitá, ale pouze v kladných hodnotách. Pro záporné hodnoty λ nejenže nalezené řešení zdaleka neodpovídá modelové funkci, ale funkce je v záporných číslech nespojitá viz graf (obr. 5.10). Proto je důležité zabránit, aby se algoritmus pro hledání kořene dostal do záporných čísel.

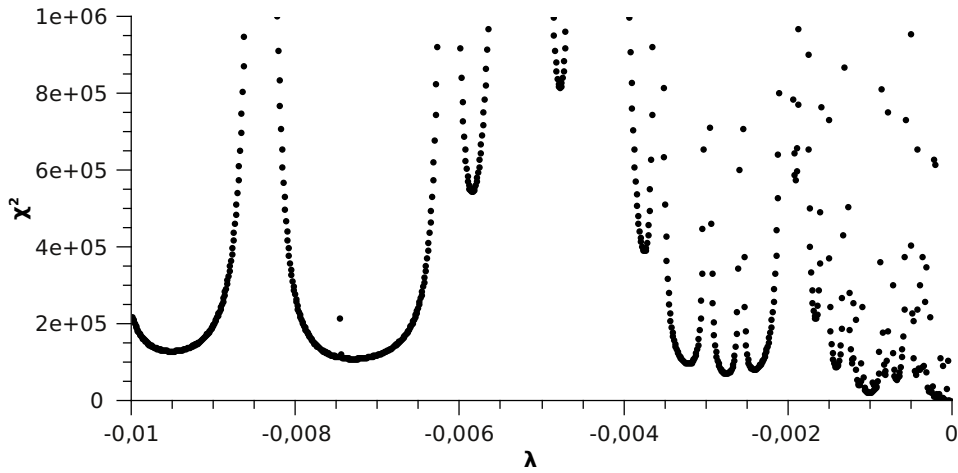
Druhý problém je samotný tvar funkce $\chi(\lambda)$. Jak je vidět na grafu (obr. 5.11b) funkce velmi prudce roste na okolí nuly a tvar se celkem výrazně liší pro různé výchozí parametry jako je očekávaná chyba, pro různé rekonstruované funkce nebo po přidání virtuálních senzorů. Největší problém je v tom, že funkce je často konkávní a pokud se tedy nehledá řešení velmi blízko kořene, tak se metoda sečen dostává do záporných čísel, což se musí řešit komplikovanými podmínkami. Druhá možnost, která se osvědčila je použít exponenciální substituci. Substituce zabrání přechodu do záporných čísel a funkce okolo kořene se vyrovná, jak je vidět na grafu (obr. 5.11a). Nevýhodou substituce je, že výsledná funkce okolo kořene má tvar exponenciály a ne přímky jako původně. Řešení tohoto problému je velmi jednoduché, stačí upravit metodu sečen pro hledání kořene pomocní exponenciál místo přímek. Výsledný tvar je velmi podobný původnímu algoritmu

$$\lambda_{i+1} = \lambda_i - \frac{\lambda_i - \lambda_{i-1}}{\ln(\chi_i^2) - \ln(\chi_{i-1}^2)} \ln(\chi_i^2)$$

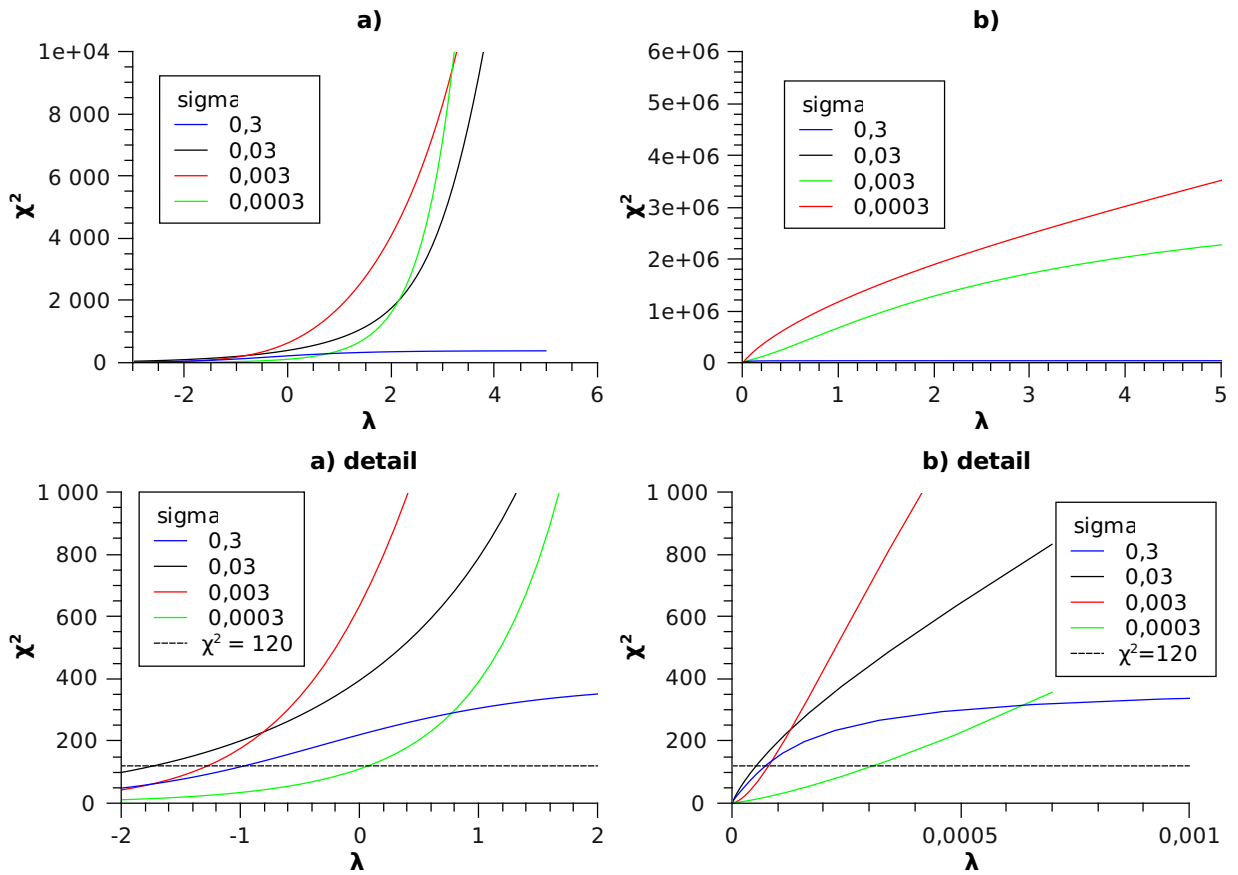
Po tomto vylepšení se zlepšila konvergence pro široké spektrum funkcí.

Dále byly v algoritmu provedené různé drobné vylepšení, například proměnou přesnost hledání řešení $|\chi^2 - 1| < \varepsilon$, protože v prvních krocích není nutná tak vysoká přesnost. Další drobné vylepšení je u virtuálního senzoru, který „vidí“ nulu mimo tokamak. Tím, že se zorné pole omezilo jenom na nejbližší okolí tokamaku, tak se hustota matice $T^T T$ snížila z 35% na 12%.

Po aplikaci všech vylepšení se rychlost rekonstrukce zvýšila ze 70 s na 2-3 s. Pro 10000 snímků by pak rekonstrukce 40×40 px na běžném počítači trvala 11 hodin a při použití nějakého výkonnějšího serveru by už doba rekonstrukce byla i přijatelná. Samozřejmě je možné snížit rozlišení rekonstrukce a protože náročnost algoritmu je



Obrázek 5.10: Typická závislost χ^2 na λ pro $\lambda < 0$



Obrázek 5.11: Závislost χ^2 na λ pro různě zvolené odchylky generovaných dat (2.5), hledané řešení je $\chi^2 = 120$. Na grafech a) je po exponenciální substituci, b) bez substituce

$O(N^6)$, tak urychlení může být značné.

Jedním z cílů bakalářské práce bylo otestovat algoritmus v GNU Octave. Základní výhodou Octave oproti MatLabu je cena, protože je zadarmo, jako největší problém

Octave se ukázala jeho nízká rychlost. Rekonstrukce 40×40 pixelů trvá v Octave 9 s, zatímco v MatLabu pouze 2-3 s. Částečně je to způsobeno tím, že Octave nedokáže využít víc než jedno jádro procesoru pro řešení matic. Rychlost rekonstrukce v MatLabu by šla ještě zvýšit použitím více procesorů. Na druhou stranu pokud je potřeba udělat velké množství rekonstrukcí, tak není problém spustit Octave vícekrát.

Algoritmus byl dále přepsán v rámci grantu do Pythonu s použitím GNU knihoven Numpy, Scipy, SciKit a Matplotlib. Rekonstrukce v Pythonu trvá 4 s a také je použito pouze jedno jádro procesoru. Výhodou oproti Octave je ale možnost využití paralelního zpracování více snímků naráz, proto ve výsledku je výkon podobný jako v MatLabu.

5.8.3 Iterační algoritmy

Další možnost optimalizace je použití iteračních algoritmů pro výpočet řešení soustavy rovnic. Výhodou by měla být teoreticky vyšší rychlost výpočtu pro řídké matice, naopak nevýhody jsou například problémy při využití více jader počítače nebo nemožnost využít iterační výpočet pro algoritmus s časovým rozlišením, protože není možné použít jako pravou stranu rovnice matici, ale jen vektor.

Bohužel řešené matice jsou velmi špatně podmíněné a iterační algoritmy s nimi mají velké problémy. Nejvíce se osvědčila metoda PCG (preconditioned conjugate gradients method), kde se jako preconditioner použil Jakobiho preconditioner [24]. Ostatní metody byly řádově pomalejší. Ale i přesto trvalo řešení jedné matice pro rozlišení 40×40 px (tzn. matice 1600×1600) PCG metodou 0,4 s, zatímco neiteračně pouze 0,2 s. Pro větší rozlišení 100×100 px byla sice iterační metoda rychlejší (8 s oproti 20 s neiteračně), ale z důvodu nižší přesnosti potřebovala iterační metoda obecně ke konvergenci více než dvojnásobek kroků, takže výsledně byla také pomalejší.

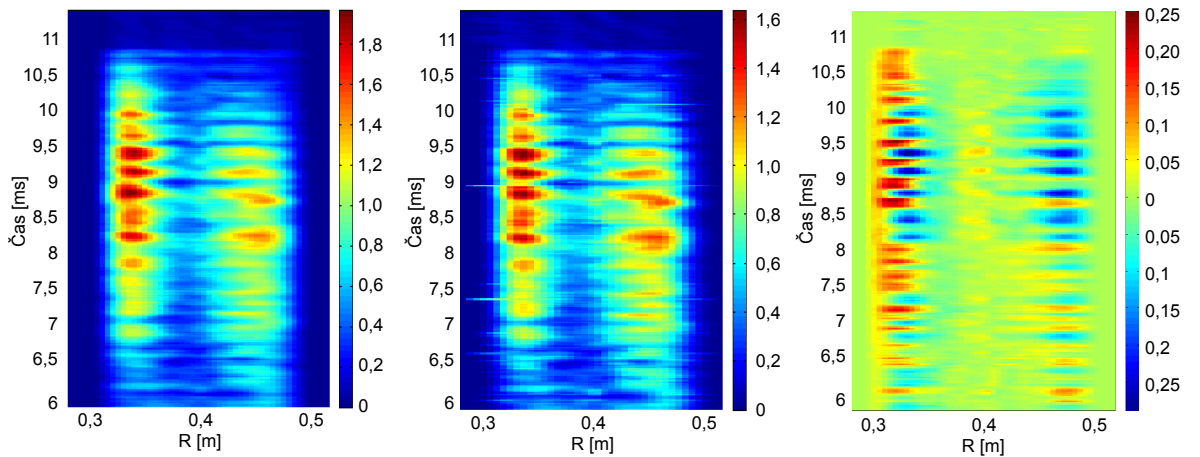
5.8.4 Počáteční podmínky

Poslední možnost, jak urychlit rekonstrukci MFI, je využití vhodného počátečního tvaru funkce g pro urychlení konvergence. Pokud se počítá více časů naráz, tak by bylo možné použít jako výchozí tvar předchozí čas. Nevýhoda tohoto přístupu je v tom, že když nezkonverguje jeden snímek, tak může poškodit i další následující snímky. Proto se více osvědčila metoda, ve které se využívá toho, že složitost algoritmu je $O(N^6)$, proto se nejdříve spočítá rekonstrukce s polovičním rozlišením, která by měla být přibližně $64 \times$ rychlejší, ta se potom interpoluje zpět na původní rozlišení a použije se jako váhová funkce. Tímto krokem se zpravidla zkrátí vnější cyklus na 1-2 iterace, tzn. lze dosáhnout až trojnásobného zrychlení. V praxi se výraznějšího zrychlení podařilo dosáhnout pouze pro případy delších výpočtů, jako například větší rozlišení ($\geq 50 \times 50$ px), hromadnou rekonstrukci nebo shlazování podél magnetického pole. Pro izotropní shlazování a rozlišení 40×40 px se rekonstrukce zrychlila pouze o 25%.

Kapitola 6

Rekonstrukce reálných dat

V době psaní této práce byl na tokamaku COMPASS funkční pouze port AU (obr. 4.2) a data z těchto detektorů byly poškozená výrazným šumem. Proto nebyla k otestování algoritmu použita data z tokamaku COMPASS, ale z menšího tokamaku Golem, které byl zatím osazen také pouze jedním detektorem. Pro rekonstrukci bylo nutné vytvořit geometrii pro tento tokamak. Jako výchozí údaje pro geometrickou matici byly použity hodnoty z práce [7]. Výsledná rekonstrukce je v grafu (obr. 6.1).

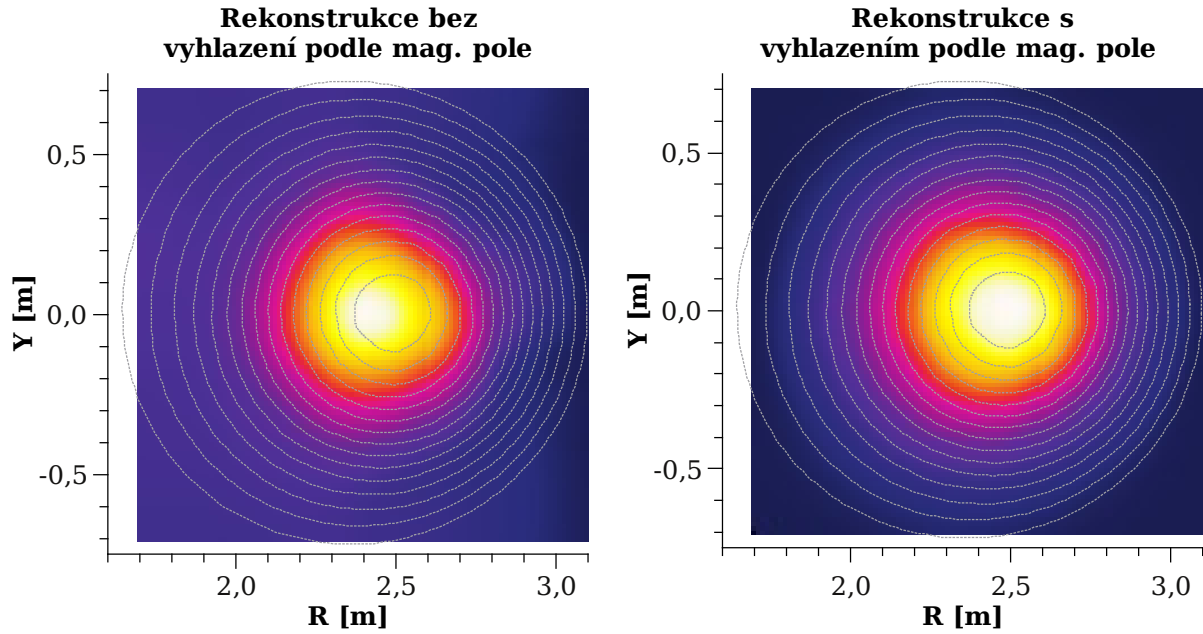


Obrázek 6.1: Rekonstrukce časového vývoje profilu výstřelu 2721, levý graf je rekonstrukce s pomocí Rapid verze (3.1), prostřední graf je obyčejná rekonstrukce pomocí MFI a v pravém grafu je rozdíl.

Při této rekonstrukci bylo ověřeno několik vlastností algoritmu. První je lepší konvergence MFI oproti vážené druhé derivaci, přesto jak je vidět na nespojitostech v prostředním grafu, v některých snímcích ani MFI algoritmus nedosáhl optimálního řešení. Dále bylo ověřeno, že použití Rapid verze nepoškodí výrazně rekonstrukci, maximální chyba dosahovala 15% s tím, že základní tvar zůstal zachován.

A na závěr se podařilo ověřit, že při rekonstrukci 272 snímků byla rapid rekonstrukce 30-krát rychlejší. Dále bylo zjištěno, že při použití více než tří vnějších cyklů při hledání MFI dochází k prudkému zhoršení hladkosti rekonstrukce na okrajích zorného pole detektorů.

Dále byla ověřena funkčnost rekonstrukce emisivity SXR z tokamaku Tore Supra (6.2). Ukázalo se, že problém u SRX rekonstrukce je velmi strmý průběh emisivity. Důsledek toho je, že řešení se podařilo najít pouze pomocí metody MFI, druhá derivace nezkonvergovala a ani metoda MFI nezkonvergovala v prvním vnějším cyklu. V druhém cyklu už metoda MFI konvergovala bez problémů. Při porovnání rekonstrukce s vyhlazováním podle magnetického pole a izotropního vyhlazování je opět vidět posunutí středu.



Obrázek 6.2: Rekonstrukce emisivity SXR u tokamaku Tore Supra. Graf vlevo je při použití izotropního shlazování a graf vpravo je při použití anizotropního shlazování podél magnetického pole

Závěr

Cílem práce bylo porovnat různé algoritmy pro tomografickou rekonstrukci a zejména se zaměřit na pixelové metody. Dalším cílem bylo vytvořit geometrickou matici pro tokamak COMPASS a ověřit chování rekonstrukce pro tuto geometrickou matici včetně různých systematických chyb a šumu v datech.

První kapitola se zabývala analytickými metodami tomografické rekonstrukce. Analytické metody mají výhodu nízké náročnosti na počítačový výkon a proto byly často používány v minulosti. Naopak jejich hlavní nevýhoda je předpoklad přesné znalosti polohy středu plazmatu a kruhové symetrie. Pokud je střed jinde než se předpokládá, tak rekonstrukce může být velmi odlišná od modelové funkce. Tento problém u testovaných pixelových metod nenastává. Pokud algoritmus zkonverguje, tak tvar rekonstrukce může být poškozený, ale základní obrysy zůstávají.

V druhé kapitole byl popsán princip pixelových metod. Pixelové metody jsou výhodné, protože počítají s konečným počtem měřených dat a jejich použití je velmi flexibilní. Je snadné přidávat další a-priori znalosti, použít různé typy vyhlazování a lze snadno započítat geometrii tokamaku. Naopak jejich nevýhodou je náročnost na počítačový výkon, protože pro rozlišení $n \times n$ pixelů je nutné vykonat přibližně n^6 operací, takže pro vyšší rozlišení roste rekonstrukční čas velmi rychle. Celkové srovnání výhod a nevýhod jednotlivých algoritmů je v tabulce (2.1).

Třetí kapitola navazuje v popisu algoritmu, který byl použit k rekonstruování funkcí v kapitole Numerické simulace. V této kapitole je stručné odvození algoritmu a popis principu. Dále je v této kapitole popis, jak se v algoritmu zabraňuje vzniku záporných hodnot rekonstrukce a jak upravit algoritmus, aby rekonstruovaná funkce byla nulová na hranici tokamaku. V sekci Časové rozlišení je popsán princip rekonstrukce více časově následujících snímků zároveň. Toho se dá využít pro zrychlení rekonstrukce.

Velmi důležitou kapitolou je Geometrie, kde je popsán způsob hledání geometrické matice a matice derivace, na kterých závisí pixelové metody. Geometrickou matici se podařilo úspěšně vytvořit pomocí algoritmu v MatLabu (A) a započítat do ní rozbíhavost chord. Také matici derivace a hlavně matici derivace podle magnetického pole se podařilo vytvořit a úspěšně otestovat. Poslední zajímavé využití předchozích algoritmů, je vytvoření geometrické matice pro kruhový profil pixelů „Abelizaci“.

Všechny tyto geometrické matice a matice derivace byly otestovány v poslední kapitole Numerické simulace. Nejdříve byla otestována kvalita rekonstrukce algoritmů s minimálním šumem. Zde dosáhnul nejlepších výsledků algoritmus hledání minima vážené druhé derivace. Zároveň se ukázalo, že vyhlazování ve směru osy x a y prefe-

ruje kosočtvercový profil, to se částečně podařilo odstranit použitím derivace podle magnetického pole. Další důležitá vec, která se podařila ověřit, je důležitost znalosti šumu v datech. Pokud šum neodpovídá vyhlazení, tak může být rekonstrukce výrazně poškozená. Kromě náhodné chyby jako je šum, může nastat i chyba systematická například posun senzoru vůči štěrbině. Pro posun štěrbin o 0,5 mm byla rekonstruovaná funkce velmi poškozená s chybou až 15% oproti rekonstrukci bez posunutí, naopak pro posun menší než 0,1 mm byla chyba zanedbatelná. V další části kapitoly byly prostudovány vlivy počátečních podmínek a nezápornosti na rekonstrukci a pouze se ověřilo, že počáteční podmínky rekonstrukci neovlivní pouze ji mohou zrychlit a dále se ověřilo, že algoritmus je poměrně odolný proti záporným hodnotám rekonstrukce.

V poslední části kapitoly Numerické simulace jsou popsány úpravy algoritmu, kterými se podařilo zvýšit rychlost rekonstrukce z 70 s na 1,5 s pro rozlišení rekonstrukce 40×40 px. Úpravami se podařilo zlepšit rychlost, ale i konvergenci. Jediný problém nastal u iteračních metod řešení matice, kde se projevila špatná podmíněnost této úlohy a algoritmus se oproti předpokladům nepodařilo zrychlit.

Celkově bych tuto práci hodnotil jako úspěšnou. Podařilo se dokončit všechny zadané úkoly a navíc se podařilo i $40 \times$ zrychlit výpočet. Dále se podařilo ověřit funkčnost matice geometrie a celého algoritmu na reálných datech z tokamaku Golem. V budoucnosti je potřeba udělat korekce na případné systematické chyby detektorů a také určit velikost očekávaného šumu v jednotlivých detektorech nebo lépe přímo korelační matici. V rámci optimalizace rychlosti výpočtu by bylo možné například implementovat řídkou síť pixelů, kdy mají jednotlivé pixely různou velikost podle jejich polohy v tokamaku.

Seznam použitých zdrojů

- [1] Jan Scheffel, Per Brunsell: *Fusion Physics - introduction to the physics behind fusion energy*, textbook, KTH Stockholm 2007
- [2] Ingesson L.C. et al.: *Tomography diagnostics: Bolometry and Soft X-ray Detection*, Fusion Science and Technology 53 (2008), 528
- [3] Anton M. et al: *X-ray tomography on the TCV tokamak*, Plasma Phys Control, Fusion 38 (1996)
- [4] Mlynář J. *Pixels Method Computer Tomography in Tokamak Experiments*, Charles University, Faculty of Mathematics and Physics, Prague, September 1995, disertační práce
- [5] Kak A.C., Slaney M. *Principles of Computerized Tomographic Imaging*, IEEE Press, 1988.
<http://www.slaney.org/pct/pct-toc.html> [cit. 1.6.2010]
- [6] Natterer F. *Numerical Methods in Tomography*, Acta Numerica, 1999
http://wwwmath1.uni-muenster.de:8000/num/Preprints/1998/natterer_1/paper.pdf [cit. 1.6.2010]
- [7] Dufková, E. *Bolometrická měření celkového vyzářeného výkonu vysokoteplotního plazmatu tokamaku CASTOR*, CVUT, 2008, bakalářská práce [cit. 1.6.2010]
<http://golem.fjfi.cvut.cz/files/students/BcTh/Dufkova.pdf>
- [8] Yasumoto Y. et al. *A New Numerical method for Asymmetrical Abel Inversion*, IEEE Transactions on Plasma Science, vol. PS-9,no.1, 1981.
- [9] Dong Y., Yi Liu, Yao L., Fu B., Chen C., Ding X. *Aplication of Soft X-ray Tomography on HL-2A*
<http://www.swip.ac.cn/kjcg/06lenw/dongyb06.pdf> [cit. 1.6.2010]
- [10] Howard J., *Tomography and reliable information*, Vol. 5, No. 7/July 1988/J. Opt. Soc. Am. A, pg. 999-1014
http://people.physics.anu.edu.au/~jnh112/AIIM/My%20publications/Howard_reliable_info.pdf [cit. 1.6.2010]
- [11] P. J. Carvalho, H. Thomsen , S. Gori, et al. *Fast tomographic methods for the tokamak ISTOK*
http://www.cfn.ist.utl.pt/17IAEATM_RUSFD/doc/files/proceedings/P13.pdf [cit. 1.6.2010]

- [12] Weisstein, Eric W. *Bessel Function of the First Kind*, From MathWorld—A Wolfram Web Resource [cit. 1.6.2010]
<http://mathworld.wolfram.com/BesselFunctionoftheFirstKind.html>
- [13] Motl L., Zahradník M. *Pěstujeme lineární algebru*, 1994 <http://www.karlin.mff.cuni.cz/~motl/mzahrad/milos.ps> [cit. 1.6.2010]
- [14] Hobza T. *Matematická statistika*, 2007
<http://people.fjfi.cvut.cz/hobzatom/mast/mast.pdf> [cit. 1.6.2010]
- [15] L.C. Ingesson *Application of natural basis functions to soft x-ray tomography*, JET Report JET-R(99)07
<http://www.iop.org/Jet/fulltext/JETR99007.pdf> [cit. 1.6.2010]
- [16] L.C. Ingesson *The Mathematics of some Tomography Algorithms used at JET*, JET Report JET-R(99)08
<http://www.iop.org/Jet/fulltext/JETR99008.pdf> [cit. 1.6.2010]
- [17] Terr, David. *Fisher Information Matrix*, From MathWorld—A Wolfram Web Resource [cit. 1.6.2010]
<http://mathworld.wolfram.com/FisherInformationMatrix.html>
- [18] *MATLAB technical documentation - mldivide, mrdivide*, The MathWorks, 2010
<http://www.mathworks.com/access/helpdesk/help/techdoc/ref/mldivide.html> [cit. 1.6.2010]
- [19] Mlynář J., Coda S., Degeling A., Duval BB.P, Hoffman F., Goodman T., Lister J.B., llobet X., Weisen H. *Investigation of the consistency of magnetic and soft x-ray plasma position measurements on TCV by means of a rapid tomographic inversion algorithm*
http://infoscience.epfl.ch/record/121278/files/lrp_732_02_hq.pdf
[cit. 1.6.2010]
- [20] *IRD Catalogue*, International Radiation Detectors, Inc.
<http://www.ird-inc.com/brochure/IRD2010.pdf> [cit. 1.6.2010]
- [21] Vácha, M. *Detection systems for measurements of high-temperature plasma radiation on the COMPASS tokamak by fast bolometers and soft X-ray detectors*, Charles University, Faculty of Mathematics and Physics, 2009, diplomová práce
- [22] Wu, Xiaolin *An efficient antialiasing technique*, 1991, Computer Graphics 25: 143–152
- [23] Weisstein, Eric W. *Method of False Position*, From MathWorld—A Wolfram Web Resource. [cit. 1.6.2010]
<http://mathworld.wolfram.com/MethodofFalsePosition.html>
- [24] Jack Dongarra *Preconditioners*, Nov 1995
http://netlib.org/linalg/html_templates/node51.html [cit. 1.6.2010]

Přílohy

Příloha A

Obsah CD

Na CD přiloženému k této práci se nacházejí algoritmy, které zde byly použity a elektronická verze této práce.

bak_prace.pdf	elektronická verze této práce
MatLab\ geometry_toresupra\ geometry_compass\ geometry_golem\ Grafy\ abelizace.m annulus.m generator_dat.m geom_mat_gen.m geom_mat_setting.m graph_derivation.m mat_deriv_B.m newtonmethod.m solve.m start_sxr.m start_sxr_abel.m sxrtminfisher_gen.m	složka s použitým algoritmem pro MatLab geometrie SXR tokamaku Tore Supra (pouze přibližná) geometrie bolometrie tokamaku COMPASS geometrie bolometrie tokamaku Golem pomocná složka pro export grafů pixelová abelizace (sekce 4.1.1) podprogram určující mezikruží v abelizaci a hranici tokamaku generátor umělých dat pro zadanou geometrickou matici výpočet geometrické matice ze zadaných virtuálních chord příprava geometrické matice, načtení geometrie, sloučení virtuálních chord skript vykreslující tvar matice derivace tvorba matice derivace podle magnetického pole pomocný skript pro hledání kořene vnitřního cyklu podprogram pro řešení soustavy $(T^T T + \lambda B^T B)g = T f$ zadání pro rekonstrukci, spojuje všechny podprogramy upravené zadání pro abelizaci hlavní algoritmus pro hledání MFI
Python\ složka s použitým algoritmem pro Python, obsah podobný jako verze pro MatLab	