České vysoké učení technické v Praze Fakulta jaderná a fyzikálně inženýrská



### Vývoj a testování metod rychlé regularizace pomocí hradlových polí pro řízení experimentu v reálném čase

DIPLOMOVÁ PRÁCE

Autor: Bc. Viktor Löffelmann Vedoucí práce: doc. RNDr. Jan Mlynář, Ph.D. Rok: 2016

# zadání

#### Prohlášení

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

Nemám závažný důvod proti použití tohoto školního díla ve smyslu § 60 Zákona č. 121/1200Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne

Viktor Löffelmann

#### Poděkování

Děkuji docentu RNDr. Janu Mlynářovi, Ph.D. za vedení mé diplomové práce, za cenné rady a připomínky, které ji obohatily.

# *Název práce:* Vývoj a testování metod rychlé regularizace pomocí hradlových polí pro řízení experimentu v reálném čase

- *Autor:* Bc. Viktor Löffelmann
- Obor: Fyzikální inženýrství
- Druh práce: Diplomová práce
- Vedoucí práce: Doc. RNDr. Jan Mlynář, Ph.D.
- Abstrakt:Tomografie měkkého rentgenového záření je nadějnou diagnostickou<br/>metodou pro tokamaky umožňující určit mimo jiné polohu plazmatu a<br/>rozložení těžkých nečistot v něm. Pro případné využití v rychlé zpětné vazbě<br/>však vyžaduje radikální optimalizaci výpočetního času. Tato práce zkoumá<br/>možnosti optimalizace tomografie založené na Tichonovově regularizaci, a to<br/>jak algoritmické, tak hardwarové zejména s využitím programovatelných<br/>hradlových polí. Navržené optimalizace jsou testovány na datech získaných z<br/>experimentů na tokamaku COMPASS. Ukazuje se, že po řadě algoritmických<br/>zjednodušení lze na současných komerčně dostupných hradlových polích<br/>počítat tomografické rekonstrukce o jedné až několika stovkách pixelů v<br/>kvalitě srovnatelné s výstupy referenčního algoritmu. Rychlost rekonstrukce<br/>je přitom dostatečná pro zpětnovazební řízení polohy plazmatu.
- Klíčová slova: Tomografie Měkké rentgenové záření Regularizace Zpětná vazba Programovatelné hradlové pole

# Title:Development and tests of rapid methods of regularisationon gate arrays for real-time control of experiments

Author: Bc. Viktor Löffelmann

- Abstract: Soft X-ray tomography in tokamaks presents a viable method for plasma position measurements as well as localization of heavy impurities. Unfortunately, it has a relatively high computational complexity which needs to be reduced significantly in order to employ tomography in real-time feedback. Several possibilities are presented to accelerate the computation using both algorithmic changes and parallel implementation with use of fieldprogrammable gate arrays. Experimental data from the COMPASS tokamak are used for testing of the optimized algorithms. The design developed is shown to be able to produce one to several hundred pixel tomographic reconstructions in real-time when implemented in contemporary hardware, with the output quality comparable to the reference tomographic method.
- Keywords: Tomography Soft X-ray Regularisation Feedback loop Field-programmable gate array

## Obsah

Úvod			
1	Tomografie plazmatu	10	
	1.1 Detekce měkkého rentgenového záření	10	
	1.2 Tomografická úloha	12	
	1.3 Algebraické metody tomografie	14	
	1.4 Metody regularizace	17	
	1.5 Algoritmus na tokamaku COMPASS	20	
	1.6 Možnosti algoritmického urychlení	22	
2	Implementace tomografie	26	
	2.1 Programovatelná hradlová pole	27	
	2.2 Řešení soustav lineárních rovnic	29	
	2.3 Gaussova eliminace	31	
	2.4 Choleského faktorizace	35	
	2.5 Jacobiho a Gaussova – Seidelova metoda	36	
	2.6 Paralelní ART	39	
	2.7 Další implementační detaily	41	
3	Testování algoritmů	44	
	3.1 Testy referenčního algoritmu	44	
	3.2 Optimalizace omezováním výpočtů	51	
	3.3 Použití iteračních algoritmů	56	
	3.4 Implementace na FPGA	61	
Záv	věr	67	
Ref	Reference		
Pří	Příloha A: Obsah CD		

## Úvod

Tokamak COMPASS, pracující od roku 2006 na Ústavu fyziky plazmatu Akademie věd České republiky, je zařízení s divertorem a plazmatem ve tvaru D. S hlavním poloměrem rovným 0.56 m jde o nejmenší z tokamaků dosahujících H-módu a majících geometrii podobnou plánovanému zařízení ITER [1]. Výzkum prováděný na COMPASSu se soustředí na fyziku okraje plazmatu a interakci plazmatu s elektromagnetickými vlnami. Vysoká variabilita stroje ovšem dovoluje i jeho využití v experimentech zaměřených například na testování nových technologií diagnostiky a řízení.

COMPASS je vybaven diagnostikami produkujícími více než 1000 výstupních signálů s rozmanitými vzorkovacími frekvencemi a objemy dat. Část z těchto signálů (některé magnetické senzory, mikrovlnný interferometr) je využita ke zpětnovazebnímu řízení těch parametrů plazmatu, které během experimentu nejsou dostatečně stabilní; Patří k nim vertikální a radiální poloha plazmatu, proud, hustota a tvarování plazmatu [2]. Tyto parametry mají různé požadavky na rychlost řízení (zejména na maximální zpoždění mezi změnou v plazmatu a odpovídající reakcí řídících systémů), a tedy i na vzorkovací frekvenci diagnostik používaných k jejich řízení a na rychlost zpracování signálu ve zpětnovazební smyčce.

Nejrychlejší zpětnou vazbu vyžaduje řízení vertikální polohy, jejíž nestabilita v tokamaku COMPASS má charakteristické trvání kolem 500 µs [3]. Tato rychlost je způsobena magnetickým tvarováním plazmatu, zejména jeho protažením ve svislém směru. Proti nestabilitě naopak působí setrvačnost plazmatu a (především) jeho elektromagnetická interakce s vodivou stěnou komory tokamaku, v níž se při pohybu plazmatu indukují vířivé proudy [4]. Tím ovšem dochází jen ke zpomalení vertikálního pohybu a zpětnovazební řízení polohy je stále nutné.

Řada diagnostik umožňuje určit polohu plazmatu; Jako vstup pro zpětnovazební řízení se ovšem nejčastěji používají senzory magnetického pole. Z jejich dat je možné rekonstruovat celý tvar magnetického pole uvnitř komory tokamaku, ale také extrahovat jen několik zvolených rysů tohoto pole za pomoci heuristických algoritmů [5]. Extrahovanými rysy jsou v našem případě hodnoty vertikální a horizontální polohy magnetické osy. Jejich výpočet je velmi rychlý, v implementaci používané na tokamaku COMPASS zabere 12 mikrosekund. K udržení plazmatu na místě postačuje provádět tento výpočet jednou za 50 µs, ačkoli ve výsledku je zpoždění mezi naměřením stavu plazmatu a vyvoláním odpovídajícího zásahu ještě o něco větší [3].

Poměrně nový přístup k detekci polohy plazmatu představuje sledování plazmatem vydávaného měkkého rentgenového záření. Tradiční a univerzální metodou jeho analýzy je tomografie, která umožňuje z dat naměřených detektory z různých směrů vypočítat rozložení emisivity záření na průřezu plazmatem. Z tohoto průřezu pak lze určit místo s nejvyšší emisivitou, ale i další vlastnosti plazmatu, jako je celkový vyzařovaný výkon, odhad teploty či přítomnost těžkých nečistot [6]. Problémem tomografických metod je jejich výpočetní náročnost, která dosud nedovoluje jejich využití v případech, kdy je nutné získat výsledek do 50 µs. Analogicky k magnetickým diagnostikám lze k rychlostní optimalizaci rentgenového

měření polohy přistupovat dvěma způsoby: významně urychlit výpočet tomografie, nebo změnit přístup k problému.

V nedávné době byly navrženy velmi rychlé tomografické algoritmy, které stále poskytují poměrně spolehlivé výsledky. Ani ty však obvykle nejsou natolik optimalizované, aby umožnily řízení vertikální polohy plazmatu v reálném čase. Vzhledem k tomu, že při radikálnější optimalizaci se může snižovat kvalita výstupu, je třeba algoritmy soustavně testovat, nejlépe s takovými vstupy, které se podobají jejich zamýšlené aplikaci. Tokamak COMPASS se zdá být vhodný k takovým experimentům, přestože jeho požadavky na rychlost zpětné vazby jsou spíše extrémní.

Mimo algoritmické optimalizace bylo navrženo implementovat tomografii na hardwaru specializovaném na digitální zpracování signálu. Pro takovýto typ úlohy jsou zejména vhodná programovatelná hradlová pole, která dovolují nízkoúrovňovou optimalizaci vstupů a výstupů a masivní paralelizaci relativně jednoduchých výpočtů. I tato zařízení však mají značná omezení, a tomografický algoritmus je pro ně nutné redukovat na co nejjednodušší funkční design. Je přitom řada možností, co omezit, a pro bezpečnou optimalizaci je opět nutné rozsáhlé testování.

## 1 Tomografie plazmatu

Plazma v tokamaku vyzařuje široké spektrum elektromagnetického záření a různé druhy částic. Záření i částice nesou informaci o řadě jevů v plazmatu nastávajících a je tedy vhodné je využívat k jejich detekci a analýze. V některých případech tato analýza vyžaduje rozsáhlé algoritmické zpracování měřených signálů, například proto, že je třeba rekonstruovat rozložení hustoty vyzařování na základě údajů detektorů s nějakým druhem prostorového rozlišení (obvykle směrových detektorů, které sledují plazma pod různými úhly).

Potřeba složitého zpracování se týká měkkého rentgenového záření, jehož rozložení vypovídá o teplotním profilu (a tudíž poloze) plazmatu a o přítomnosti a transportu nečistot v něm [6]. Jiným příkladem je neutronové záření, které je užitečné při mapování hustoty fúzního výkonu (protože zaznamenávané neutrony pocházejí z fúzních reakcí), ale i iontového teplotního profilu. U neutronů navíc může jít nejen o prostorovou rekonstrukci, ale také o rozlišení jejich energetického spektra, jež některé neutronové detektory zaznamenávají, ale ne v přímo použitelné podobě [7].

Žádaných výsledků lze v obou zmíněných případech dosáhnout řešením algebraických inverzních úloh známých jako *tomografie* (respektive podobné úlohy zvané *unfolding* pro rekonstrukci energetického spektra). Tyto metody jsou v současnosti na tokamacích používány i pro diagnostiky zaznamenávající viditelné, tvrdé rentgenové či gama záření. Na tokamaku COMPASS lze tomografii provádět kromě měkkého rentgenového záření také pro celkový vyzářený výkon (tj. bolometrickou diagnostiku).

#### 1.1 Detekce měkkého rentgenového záření

Měkké rentgenové záření vzniká v plazmatu několika mechanismy. Jedním z nich jsou srážky elektronů s ionty, při kterých je elektronům uděleno velké zrychlení, a ty pak vyzařují brzdné záření. Jeho intenzita silně (exponenciálně) závisí na teplotě, což činí měkkou rentgenovou část spektra vhodnou k rozpoznání nejteplejší oblasti plazmatu. Předpokládáme-li pak, že místo s nejvyšší teplotou (a tlakem) se nachází na magnetické ose plazmatu, můžeme použít tuto diagnostiku k určení polohy magnetické osy a tedy "polohy plazmatu". Podobné využití nachází měkké rentgenové záření při sledování nestabilit, které lokálně mění teplotu, jako jsou magnetické ostrovy nebo pilová nestabilita [6].

Brzdné záření ovšem závisí i na efektivním náboji, který je vysoký, pokud jsou v plazmatu přítomny těžké nečistoty. Nečistoty navíc přispívají k emisivitě měkkého rentgenového záření i jinými mechanismy než brzdným zářením, totiž zářivou rekombinací a charakteristickým zářením, které fungují díky tomu, že dostatečně těžký iont si udrží část elektronového obalu i v podmínkách tokamaku. Je-li těžkých nečistot dost, jimi způsobené záření je silnější než brzdné záření vznikající na vodíkových iontech [8]. To maří měření polohy, vzhledem k tomu, že nečistoty se mohou koncentrovat i jinde než ve středu plazmatu. Zároveň to však dovoluje použít měkkou rentgenovou diagnostiku k lokalizaci nečistot, případně i k detailnější analýze jejich zastoupení [9].

Pro zjednodušení systému měkké rentgenové diagnostiky lze předpokládat, že plazma v

tokamaku je toroidálně symetrické. Pak stačí umístit detektory v jedné toroidální poloze, a použít detektory sledující jen tenkou vrstvu plazmatu podél poloidálního průřezu zařízením. V takovém případě nelze sledovat například toroidální módové číslo ostrovů a jiné jevy, které nesplňují předpoklad symetrie; Strukturu plazmatu na průřezu v dané toroidální poloze ovšem odvodit lze.

V případě nedostatku detektorů je možné zavést další zjednodušující předpoklady o plazmatu. To je v analýze záření plazmatu často využívaná technika, protože rozmístění detektorů bývá z konstrukčních důvodů omezené na několik míst okolo vakuové komory. Vzhledem k vlastnostem plazmatu lze například očekávat, že profil měkké rentgenové emisivity bude hladký. Také je možné využít znalosti tvaru ploch konstantního magnetického toku (k tomu je potřeba doplnit rentgenové detektory magnetickými diagnostikami) a předpokládat, že profil emisivity je zvláště hladký ve směru rovnoběžném s nimi [6]. V extrémním případě se dá počítat s tím, že podél ploch konstantního magnetického toku je emisivita konstantní. Potom lze získat zajímavé výsledky (například celkový vyzářený výkon) i s jediným prostorově rozlišeným senzorem; Je ovšem zřejmé, že takovým omezením přijdeme o informaci o jevech, jako je rozložení nečistot.

Jedním ze způsobů, jak se širokoúhlými detektory dosáhnout prostorového rozlišení, je umístit detektory do dírkové komory. V takovém uspořádání je mezi detektory a sledovaným objektem umístěna přepážka z materiálu stínícího rentgenové záření s malým otvorem. Detektor v daném místě za přepážkou pak zaznamenává pouze záření ze směru, ve kterém leží otvor. Pro získání prostorového rozlišení v jednom směru pak stačí do dírkové komory vložit řadu detektorů orientovanou ve směru požadovaného rozlišení. Tato konstrukce je znázorněna na obrázku 1.1. Pokud je požadováno rozlišení jen v jednom směru (*řádková kamera*), může mít dírka v přepážce protáhlý tvar s delší stranou kolmou na řadu detektorů. K detektorům pak projde víc záření, aniž by došlo ke ztrátě rozlišení v žádaném směru (dojde ovšem ke zhoršení kolimace detekovaných paprsků ve druhém směru). Analogicky je možné dosáhnout i dvojrozměrného rozlišení.



Obr. 1.1: Dvě zobrazení objektu řádkovou dírkovou komorou se dvěma detektory v řádku. Uspořádání detektorů je v obou případech stejné, liší se velikost dírky.

Jako detektory pro měkké rentgenové záření jsou často používány polovodičové nebo Schottkyho fotodiody [6]. Tyto diody jsou zapojovány v závěrném směru, takže bez osvětlení jimi teče jen malý proud, a v blízkosti jejich hradlové vrstvy se objeví oblast, v níž téměř chybí volné nosiče náboje. Tato oblast je pak citlivá na fotoelektrický jev, který volné nosiče náboje vytváří. Díky tomu dojde ke zvýšení proudu protékajícího diodou (při konstantním napětí) při jejím vystavení záření s dostatečnou energií. Mezní energie používaných diod je přitom tak nízká, že tyto diody detekují i ultrafialové záření; To přitom u tokamakového plazmatu převažuje [8]. Pro účely detekce měkkého rentgenového záření se tudíž před diody umisťuje filtr, který ultrafialové záření nepropustí. Na tokamaku COMPASS je použit tento druh detektorů doplněný o beryliový filtr tlustý 10 µm [10]; Podobné detektory, avšak bez filtru, jsou užívány jako bolometry, tedy pro měření celkového toku plazmatem vyzařovaného elektromagnetického záření.

Bohužel se ukazuje, že polovodičové fotodiody jsou náchylné k radiačnímu poškození, zejména při ozáření neutrony, k němuž bude docházet v tokamacích pracujících s deuteriem a tritiem. Proto se počítá i s využitím jiných typů detektorů neobsahujících polovodiče, jako jsou vakuové fotodiody nebo *Gas Electron Multipliers* (GEM). Vakuové fotodiody jsou také založeny na fotoelektrickém jevu, ovšem v tomto případě k němu dochází na povrchu kovové katody. Podobně jako polovodičové detektory jsou vakuové fotodiody citlivé na široké energetické spektrum záření, byť jejich absolutní citlivost je nižší [11]. Detektory typu GEM využívají fotoefekt pouze k počáteční konverzi fotonu na elektron. Signál je dále zesílen lavinovým efektem při průchodu elektrickým polem mezi dvěma elektrodami (nebo několika páry elektrod). Elektrody v párech jsou umístěny blízko sebe (jsou realizovány jako oboustranně pokovená plastová fólie, s otvory pro průchod elektronů), takže nezhoršují prostorové rozlišení. Zesílený signál je pak detekován jako oddělené proudové impulzy, přičemž z tvaru těchto impulzů je možné získat i informaci o energii detekovaného záření [8].

#### 1.2 Tomografická úloha

Měkké rentgenové i ostatní zmíněná záření snadno procházejí zkoumaným objektem (objekt je *opticky tenký*), což je dobrý výchozí bod pro pozorování vnitřní struktury objektu. Samotný obraz zaznamenávaný detektory však neukazuje přímo tuto (obecně trojrozměrnou) strukturu, nýbrž jen její projekce, v nichž se bližší a vzdálenější části objektu překrývají. V určitých případech lze z projekcí rekonstruovat lokální informaci o vnitřku objektu, v ideálním případě pak obraz, který spíše než projekci odpovídá průřezu objektem. Metody tomografie slouží právě k takovém zpracování projekcí.

Tomografie má široké uplatnění v medicíně, ale i v průmyslu, výzkumu a vývoji, geologii, archeologii či astrofyzice. Jako médium přinášející informaci o vnitřku objektu může sloužit záření produkované objektem (tomu se někdy říká *emisní tomografie*), záření z vnějšího zdroje tlumící se v objektu (*transmisní tomografie*), odražené záření, ale i například elektrický proud v prostředí s proměnným odporem. Pro zmíněné aplikace byla vyvinuta řada metod přizpůsobených jejich různorodým podmínkám. Ve většině případů se volí strategie zaznamenání velkého množství projekcí z mnoha směrů, což umožňuje rekonstruovat trojrozměrný obraz s velkým rozlišením a značně věrný předloze. Někdy je ovšem množství měřených dat omezeno, například snahou o co nejmenší ozáření pacienta (pozitronová emisní tomografie, SPECT [12]), kvůli principiálním nedostatkům používaného média (seismologie

[13], mionová tomografie [14]) nebo kvůli omezením kladeným na konstrukci či montáž detektorů (což je případ tomografie plazmatu). Pak je hlavním úkolem rekonstrukčních metod podat spolehlivé výsledky, byť s menším množstvím detailů.

Téměř všechny tomografické metody využívají k rekonstrukci počítačový algoritmus, který vychází z matematického popisu transformace struktury objektu do projekcí a implementuje některý z postupů umožňujících tuto transformaci invertovat. Proto se pro ně někdy používá termín *výpočetní tomografie* (za účelem odlišení od metody známé jako *konvenční tomografie*, v níž se používá dlouhá expozice spojená s pohybem aparatury, který ve výsledném obrazu rozmaže vše kromě zvolené roviny ostrosti [15]).

První metody vyvinuté pro výpočetní tomografii se podobaly současné tomografii plazmatu v tom ohledu, že umožňovaly rekonstruovat dvojrozměrný obraz z jednorozměrných projekcí, tedy získat jeden řez objektem. Projekci modelovaly s pomocí *Radonovy transformace*, značně idealizovaného předpisu počítajícího s pořízením projekce z každého z nekonečně mnoha směrů a s nekonečně jemným rozlišením u každé projekce. Radonova transformace převádí funkci *f* prostorových proměnných *x* a *y* na funkci v proměnných *p* a  $\theta$ , které mají význam parametrů přímky, podél které se původní funkce integruje:

$$R[f](p,\theta) = \int f(p\cos(\theta) + t\sin(\theta), p\sin(\theta) - t\cos(\theta)) dt$$
(1.1)

Výhodou takto zjednodušeného modelu projekce je skutečnost, že k němu lze odvodit analytický vzorec pro inverzní transformaci. Klíčovým bodem odvození je vztah známý jako *Fourier slice theorem*, který říká, že jednorozměrná Fourierova transformace projekce (tj. funkce  $R[f](_, \theta)$  pro zvolené  $\theta$ ) odpovídá jistému řezu dvojrozměrnou Fourierovou transformací původního obrazu [16]. Lze tedy vypočítat Fourierovy transformace všech projekcí, namapovat získané hodnoty do dvojrozměrné frekvenční oblasti a odtud provést inverzní dvojrozměrnou Fourierovu transformaci, čímž získáme rekonstruovaný obraz. V implementaci posledního kroku je přitom třeba počítat s tím, že frekvenční oblast není známými hodnotami pokryta rovnoměrně; Nízké frekvence jsou pokryty lépe než vysoké. V numerické integraci nutné pro výpočet inverzní Fourierovy transformace tak musí vysokofrekvenční vzorky dostat vyšší váhu. Přesnost výsledku ovšem i pak závisí na tom, jak hustě je frekvenční oblast navzorkována.

Popsaný postup s počítáním Fourierových transformací a jejich inverzí má v přímočaré implementaci problémy s numerickou stabilitou, tj. zvýrazňuje nepřesnosti přítomné v datech [17]; Lze ho však snadno obejít. Nejprve se s využitím linearity Fourierovy transformace zamění pořadí operací: Vypočte se jednorozměrná Fourierova transformace zvolené projekce, výsledek se vynásobí vahami, provede se dvojrozměrná inverzní Fourierova transformace, a nakonec se sečtou obrázky takto získané ze všech projekcí. Úvodní proces se dvěma druhy Fourierovy transformace proloženými vážením odpovídá filtrování projekcí a jejich následnému roztažení přes definiční obor rekonstrukce, což lze provést i bez použití Fourierových transformací. Takto implementovaný algoritmus je znám jako *filtrovaná zpětná projekce (filtered back-projection)* [16].

Ačkoli je filtrovaná zpětná projekce odvozena jako analyticky přesná metoda, ve své prakticky použitelné podobě trpí řadou nepřesností. První problém plyne již z toho, že proces projekce neodpovídá ideální Radonově transformaci, nýbrž jen nějakému jejímu diskrétnímu přiblížení

(kvůli tomu není zaručena ani obecná možnost provést přesnou rekonstrukci, nezávisle na tom, jaký algoritmus použijeme). Stejně diskretizovaný musí být i inverzní proces, zejména filtrace. Proto se v rekonstrukcích získaných filtrovanou zpětnou projekcí objevují typické *artefakty* (systematické chyby), a to tím výraznější, čím méně projekčních dat bylo použito [16, 18].

Filtrovaná zpětná projekce se od svého uvedení dočkala postupného rozšíření na trojrozměrné tomografické problémy a dosud je nejpoužívanějším algoritmem v lékařské a průmyslové tomografii [19]. Její implementace se vyznačují vysokou rychlostí a při této rychlosti i dobrou přesností. Do oborů, jako je tomografie plazmatu, se ovšem nerozšířila, a to zřejmě kvůli svým vysokým požadavkům na vstupní data. Pro tyto účely byla vyvinuta řada odlišných algoritmů, které k tomografické úloze přistupují více či méně jako k řešení obecného inverzního problému.

#### 1.3 Algebraické metody tomografie

Algoritmy z rodiny filtrované zpětné projekce počítají s tím, že projekční data pokrývají parametrický prostor (v parametrech p,  $\theta$  i v odpovídající frekvenční oblasti) tak hustě, aby se přes něj dalo s rozumnou přesností numericky integrovat, a navíc v nějakém smyslu pravidelně, aby se snadno implementovala filtrace. V některých aplikacích tomografie jsou ovšem projekční data dostupná jen v určitém rozmezí směrů, nebo jen v několika konkrétních směrech určených uspořádáním detektorů (z obrázku 1.2 je patrné, že se to týká tomografie plazmatu). Předpoklady filtrované zpětné projekce lze ovšem porušit i jinak, například tím, že v objektu dochází k ohybu paprsků nebo (u emisní tomografie) k reabsorpci záření. Někdy je vhodné počítat s tím, že jednotlivé detektory neznamenávají záření integrované přes přímku, ale přes pruh (nebo dokonce prostorový útvar) o nenulové, případně proměnné, šířce (přes *chordu*). V takovýchto případech se lépe než filtrovaná zpětná projekce uplatní tomografické algoritmy známé jako *algebraické*.



Obr. 1.2: Uspořádání detektorů měkkého rentgenového záření na tokamaku COMPASS a jemu odpovídající vzorkování funkce v proměnných p, θ. Šedé kontinuum představuje Radonův obraz vnitřku vakuové komory.

Algebraické metody vycházejí z obecného popisu projekce, tedy ne nutně z nějakého přiblížení Radonovy transformace. Místo nekonečného počtu projekcí s dokonalým rozlišením od začátku počítají s diskrétním uspořádáním detektorů, což dovoluje detailně popsat, který detektor registruje jakou část rekonstruovaného objemu. Podobně je nutné diskretizovat rekonstruovaný obraz, a to tak, že jej vyjádříme jako lineární kombinaci zvolených bázových funkcí. Těchto funkcí musí být konečný počet, čímž je omezen prostor možných výsledků. Úloha modifikovaná diskretizací vstupu i výstupu má podobu konečné soustavy rovnic pro konečně mnoho neznámých (pro každou bázovou funkci je třeba určit jeden neznámý koeficient) a dá se řešit některou z metod určených pro takové problémy. Rovnice navíc typicky bývají lineární, což je dáno lineární odezvou detektorů na jas objektu (výše zmíněný případ s ohybem paprsků je ovšem výjimka [16]).

Velikost použité báze ovlivňuje nejen variabilitu možných řešení, ale i náročnost a řešitelnost diskretizované úlohy. Při malém počtu bázových funkcí hrozí, že úloha nebude mít řešení (ani skutečná struktura objektu nemusí úlohu řešit kvůli nepřesnostem v její formulaci a chybám v datech); Pak lze nalézt alespoň přibližné řešení, například pomocí metody nejmenších čtverců nebo obecnějšími statistickými metodami založenými na principu maximální věrohodnosti (tyto metody jsou známé jako *expectation maximization*). V opačném případě řešení může existovat, ale není jednoznačné. Lze si pomoci rozšířením projekčních dat nebo doplněním úlohy o další podmínky kladené na řešení, ovšem výpočetní náročnost se tím nejspíš ještě zvýší. Je tedy dobré zvolit takovou bázi, která umožní vyjádřit co nejvíce zajímavých rysů tomografické rekonstrukce při co nejnižším počtu bázových funkcí.

Jedním z prvních, kdo tento problém řešili, byl Allan Cormack (Nobelova cena 1979 [20]), který jako bázi použil Zernikovy polynomy (do zvoleného řádu). To jsou funkce definované na jednotkovém kruhu, které lze vyjádřit jako součin Čebyševova polynomu v radiální souřadnici a funkce sinus nebo kosinus v úhlové souřadnici škálované tak, aby zde měla celočíselný počet period. Jejich význačnými vlastnostmi jsou ortogonalita na daném definičním oboru a fakt, že jejich Radonova transformace (a tudíž i jí odpovídající projekční matice) se dá analyticky spočítat [21]. Cormackova metoda byla původně zamýšlena pro medicínu, ale našla uplatnění i v tomografii plazmatu, kde se ukázala její schopnost rekonstruovat objekt s určitým typem symetrie (v tomto případě šlo o magnetické ostrovy v plazmatu s kruhovým průřezem [22]).

Problémem Cormackovy metody je špatná schopnost rekonstruovat objekty, které zmíněnou symetrii nemají. Pro takové případy je třeba použít vyšší počet bázových funkcí (tj. Zernikovy polynomy vyššího řádu), jinak dochází k aliasingu způsobenému tím, že ve spektru řady Zernikových polynomů se vyskytují nízkofrekvenční i vysokofrekvenční složky. Porušení symetrie je bohužel časté i u tokamaků, jednak proto, že moderní tokamaky nemívají kruhový průřez plazmatu [22], a dále proto, že občas předem neznáme polohu plazmatu a tím střed jeho symetrie. Cormackovu metodu lze pro tyto případy zobecnit, přesto se v současnosti častěji setkáme s jinými typy bázových funkcí, zejména s *pixely*.

Pixely v tomografickém kontextu jsou funkce mající konstantní hodnotu na nějaké oblasti (označované také slovem pixel) a nulové všude jinde. Báze jimi tvořená je sestavena tak, aby každý bod definičního oboru byl pokryt právě jedním pixelem. Obvyklá je pravidelná síť čtvercových či obdélníkových pixelů, ale ve speciálních případech se lze setkat i s jinými tvary, například se soustřednými mezikružími pro rekonstrukce objektů, u nichž se předpokládá

kruhová symetrie. Síť obdélníkových pixelů umožňuje rekonstruovat rozmanité objekty, je jich ovšem pro podobné prostorové rozlišení třeba více než bázových funkcí využívajících symetrií objektu. Na druhou stranu, projekční matice pro takovouto bázi je řídká, protože do zorného pole jednotlivých detektorů vždy zasahuje jen málo pixelů; To umožňuje zavést do výpočtu jisté optimalizace.

Modifikací koncepce pixelů je interpolativní báze [22], skládající se ze spojitých funkcí nenulových na omezené oblasti a částečně se překrývajících se sousedními funkcemi; Každá z těchto funkcí nabývá v nějakém bodě hodnoty 1, zatímco ostatní bázové funkce jsou tam nulové. Lineární kombinace z takovéto báze je také spojitá a má význam funkce interpolované mezi význačnými body jednotlivých prvků báze (koeficienty lineární kombinace interpolativní báze odpovídají hodnotám ve význačných bodech, zatímco u pixelů má funkce tuto hodnotu na celé ploše příslušného pixelu). Podobně jako pixely je takováto báze poměrně obecná, využívá jen předpokladu spojitosti (nebo nějakého typu hladkosti, podle zvoleného způsobu interpolace) rekonstruovaného profilu. Odpovídá-li tento předpoklad skutečnosti, je k rekonstrukci profilu s danou přesností potřeba méně bázových funkcí než při použití pixelů [22]. Obrázek 1.3 ukazuje možnou podobu interpolativního bázového prvku pro dvojrozměrnou tomografii.

Další otázkou je, jak výše definovanou soustavu lineárních rovnic vyřešit. Je-li malá (málo projekčních dat a malé rozlišení rekonstrukce), lze ji řešit některou z přímých metod, jejichž typickým zástupcem je Gaussova eliminace [23]. Složitost této a podobných metod ovšem roste lineárně s počtem neznámých a s druhou mocninou počtu rovnic. Obvyklejším případem je tak použití iteračních algoritmů a hledání řešení s omezenou přesností. Výhodou řady iteračních algoritmů je i fakt, že nepracují s celou maticí najednou, ale například jen s jedním řádkem v každém kroku. Díky tomu se nemusí uchovávat v paměti celá matice, lze místo toho sestavit každý řádek až když je třeba. To umožňuje řešit s pomocí algebraických metod i velmi velké úlohy (jako jsou ty trojrozměrné) s omezenou operační pamětí.

Oblíbeným iteračním algoritmem používaným k tomuto účelu je algoritmus Kaczmarzův, který v každém kroku upraví průběžné řešení tak, aby přesně splňovalo jednu rovnici ze soustavy [16]. Tato úprava je přitom z mnoha možností provedena tak, aby změna provedená v řešení byla v eukleidovské normě co nejmenší; Jestliže jedna lineární rovnice reprezentuje nadrovinu v prostoru možných řešení, jde o kolmé promítnutí řešení na tuto nadrovinu. Tímto způsobem se na řešení postupně aplikují všechny rovnice ve zvoleném pořadí, a to obvykle vícekrát za



Obr. 1.3: Interpolativní pixely odpovídající bilineární interpolaci mezi body ve čtvercové síti; Hodnota lineární kombinace v každém uzlu sítě odpovídá amplitudě interpolativního pixelu se středem v tomto uzlu.

sebou, ačkoli v některých případech se dá dosáhnout optimálních výsledků po jedné iteraci [24]. Kromě toho platí, že algoritmus konverguje vždy, když existuje řešení, a pokud neexistuje, umožňuje postup najít alespoň jeho přiblžení. Na různých variacích Kaczmarzova algoritmu je postavena tomografická metoda ART (Algebraic Reconstruction Technique) a řada dalších podobných metod s názvy jako SIRT (Simultaneous Iterative Reconstruction Technique) nebo SART (Simultaneous Algebraic Reconstruction Technique) [16, 25].

Kaczmarzův algoritmus implicitně využívá principu, s jehož pomocí lze dosáhnout přesného výsledku opakovanou aplikací přibližné metody. Jde o to, že po každém kroku vypočteme z mezivýsledku reziduum (tj. přímou aplikací projekční matice získáme projekce odpovídající mezivýsledku a odečteme je od skutečně naměřených projekcí), a v příštím kroku dosadíme toto reziduum na místo pravé strany úlohy. Na podobném principu (více či méně zřetelném) je založena řada iterativních metod pro tomografii. Výběrem přibližného inverzního algoritmu můžeme měnit strategii iterační metody: Rychlý a málo přesný algoritmus vede k metodě s mnoha krátkými iteracemi a naopak. Příkladem prvního přístupu je Landweberova metoda, u níž přibližné řešení spočívá v násobení pravé strany transponovanou projekční maticí (čemuž by se dalo říkat nefiltrovaná zpětná projekce) [18]. Opačný extrém představují metody, v nichž je "přibližné řešení" získáváno filtrovanou zpětnou projekcí nebo jí podobným algoritmem. Iterace pak slouží k opravě chyb pocházejících z diskrétního přiblížení inverzní Radonovy transformace nebo jiných efektů, které byly v obyčejně přesné filtrované zpětné projekci zanedbány (například nenulová šířka chord nebo rozbíhavost ve dvou směrech u trojrozměrné tomografie [18]).

#### 1.4 Metody regularizace

Všechny metody, které mají ambice získat z projekcí přesnou rekonstrukci obrazu, jsou v ideálním případě (pomineme-li diskretizační chyby, široké chordy či konečnou numerickou přesnost počítačů) ekvivalentní filtrované zpětné projekci. Jejího analytického předpisu pak lze využít k odvození závěrů týkajících se všech těchto metod. Významným závěrem je jejich vysoká citlivost na šum přítomný v datech, která plyne z vážení naměřených hodnot ve frekvenční oblasti [18]. Toto vážení má za cíl vrátit původní význam vysokým frekvencím, které jsou mechanismem projekce utlumeny. Pokud se však v projekcích vyskytuje bílý šum, jeho vysokofrekvenční složky jsou tímto vážením zvýrazněny, aniž by předtím prošly nějakým útlumem (předpokládáme, že šum vzniká na detektoru nebo při následném analogovém zpracování, případně i při digitalizaci; V některých aplikacích ovšem dominují jiné druhy šumu, jako je šum způsobený rozptýlenými fotony [26]). Tomografická rekonstrukce je tedy příkladem *špatně podmíněné úlohy* (což obecně znamená, že malá odchylka ve vstupních datech může vyvolat velkou změnu na výstupu).

Pro kvalitní rekonstrukci je tedy třeba omezit šum v projekčních datech. Toho lze dosáhnout vyšší expozicí, která snižuje kvantový šum (způsobený tím, že záření přichází do detektoru po jednotlivých fotonech, přičemž počet fotonů za jednotku času má Poissonovo rozdělení) i šum vznikající v následném zpracování (je třeba nižší zesílení). Ukazuje se také, že při dané expozici lze optimalizovat šumové poměry v rekonstrukcích vhodným rozdělením expozice mezi projekce [27]. Ne vždy ovšem situace dovoluje takovéto postupy. V lékařské tomografii je expozice omezena dávkou přijatou pacientem; V tomografii plazmatu je podobný problém s požadavkem na rychlé vzorkování v čase, a počet detektorů (a tím i projekcí) bývá poněkud

nižší než zmíněné optimum.

I z řídkých a relativně silně zašuměných projekcí je ovšem možné získat použitelnou rekonstrukci, pokud se vzdáme myšlenky přesného a univerzálního algoritmu. Metody, které to umožňují, jsou známé pod názvem regularizace. Jako nejjednodušší se jeví možnost upravit (odšumovat) data vstupující do algoritmu, nebo naopak obrázek na výstupu. Tento přístup však má značná omezení. Projekční data v tomografii plazmatu jsou řídká a není snadné rozpoznat a odstranit jejich šumovou složku; Naopak přitom hrozí zavedení nových, systematických chyb do těchto dat a těžko předvídatelné znehodnocení rekonstrukce. Ve stejném případě nelze použít ani druhou metodu, protože šum v rekonstrukci je značně zvýrazněn, případně má poškození spíše charakter artefaktů. Naštěstí se dají použít univerzálnější regularizační metody, které místo vstupu a výstupu modifikují celý průběh algoritmu.

Regularizační metody často využívají toho, že o objektu něco víme i bez znalosti projekcí (máme *a-priori informaci*). Například o poloidálním profilu rentgenové emisivity plazmatu můžeme předpokládat, že je hladkou funkcí prostorových souřadnic, případně že profil emisivity je hladší ve směru rovnoběžném s plochami konstantního magnetického toku než ve směru kolmém. Takovýto druh informace dovoluje snížit obecnost tomografické metody a v jejím rámci pak lépe využít naměřená data.

Poměrně jednoduchou variantou regularizace je pečlivý výběr bázových funkcí, a s tím související omezení jejich počtu. Zároveň dojde k omezení prostoru přípustných řešení, což je v tomto případě žádoucí. Lineární prostor řešení se však snížením počtu bázových funkcí může změnit jen na jiný lineární prostor, což nemusí odpovídat dostupné a-priori informaci (to je případ výše zmíněné Cormackovy metody). Navíc se ukazuje, že výsledné řešení může být na výběr bázových funkcí velmi citlivé [6].

Obvyklejší možností je tudíž zvolit obecné bázové funkce (dostatečně malé pixely) a a-priori informaci do úlohy zavést uměle. Díky této a-priori informaci je možné z původně nedostatečně určené a špatně podmíněné úlohy udělat úlohu s jednoznačným řešením (proto "regularizace"), které je málo citlivé na změny v datech. Jednou z možností, jak to provést, je Tichonovova regularizace [28]. Ta vyžaduje a-priori informaci v podobě *hodnotícího funkcionálu*, který každému možnému řešení přiřazuje číselnou hodnotu; Čím je tato hodnota nižší, tím vyšší pravděpodobnost se přisuzuje výskytu daného řešení (tj. u profilu emisivity plazmatu bude tím vyšší, čím je profil hladší). Do podoby minimalizace funkcionálu se upraví i původní tomografická úloha, a následně se hledá řešení minimalizující jistou kombinaci těchto dvou funkcionálů. Zapíšeme-li původní úlohu jako řešení soustavy

$$\mathbf{T} \, \boldsymbol{g} \; = \; \boldsymbol{f} \; , \tag{1.2}$$

kde vektor g je složen z (neznámých) koeficientů lineární kombinace bázových funkcí, f jsou údaje zaznamenané detektory a projekční matice **T** popisuje vztah mezi nimi, pak regularizovanou úlohu lze zapsat jako

$$\underset{\boldsymbol{g}}{\arg\min} \|\mathbf{T}\boldsymbol{g} - \boldsymbol{f}\|^2 + \lambda O(\boldsymbol{g})$$
(1.3)

kde  $O(\boldsymbol{g})$ značí hodnotící funkcionál. Parametr $\lambda$  (regularizační parametr) byl zaveden pro

snadné vyvážení vlivu obou členů, ideálně tak, aby se řešení úlohy příliš neodchylovalo ani v jednom z nich. Je zjevné, že pokud byla původní úloha pouze nedostatečně určená (a ne špatně podmíněná), řešení nové úlohy se i při libovolně malé hodnotě  $\lambda$  přikloní ke tvaru co nejlépe splňujícímu předpoklady vyjádřené funkcionálem *O*. Větší péči je třeba výběru  $\lambda$  věnovat u úloh špatně podmíněných, u nichž lze očekávat, že za cenu mírného zvýšení hodnoty prvního členu ve vztahu (1.3) bude nalezeno řešení vyhovující a-priori informaci významně lépe. Bohužel není vždy jasné, jak tento parametr zvolit, ačkoli pro to byla vymyšlena řada postupů.

Tichonovova regularizace ve výše uvedené podobě předpokládá, že řešená úloha je tak malá, abychom si mohli dovolit manipulovat s celou projekční maticí najednou, respektive abychom dokázali v rozumném čase vyřešit minimalizační úlohu obsahující násobení touto maticí. Podobného efektu se ovšem dá dosáhnout i při použití iteračních tomografických metod jako je ART, které špatnými možnostmi optimalizace netrpí. Za tím účelem je nutné modifikovat iterační krok algoritmu tak, aby snižoval nejen reziduum průběžného řešení, ale i hodnotu hodnotícího funkcionálu na něm, opět ve vhodném poměru. Metodu SIRT (což je Kaczmarzův algoritmus s výpočtem rezidua jen jednou za *n* kroků, kde *n* je počet rovnic [16]) lze odvodit tak, že požadujeme, aby po každém iteračním kroku přesně platil jeden vybraný řádek vztahu (1.2); Pro jednoznačnost postupu předpokládáme, že oprava řešení v daném kroku je násobkem vybraného řádku projekční matice. Iterační krok výsledné metody má tvar

$$\boldsymbol{g}^{k+1} = \boldsymbol{g}^{k} + \omega \left( \boldsymbol{f}_{i} - \boldsymbol{T}_{i} \boldsymbol{g}^{\lfloor k/n \rfloor \cdot n} \right) \frac{\boldsymbol{T}_{i}}{\left\| \boldsymbol{T}_{i} \right\|^{2}}, \qquad (1.4)$$

kde  $T_i$  je onen vybraný řádek projekční matice a  $\omega$  je relaxační parametr ovlivňující výskyt a rychlost konvergence. Analogický postup můžeme (různými způsoby) aplikovat na vztah (1.3). Předpokládáme-li opět, že neregularizovaná složka opravy má mít směr shodný s jedním řádkem matice, získáme pro iterační krok vzorec

$$\boldsymbol{g}^{k+1} = \boldsymbol{g}^{k} + \omega \left( \boldsymbol{f}_{i} - \boldsymbol{T}_{i} \boldsymbol{g}^{\lfloor k/n \rfloor - n} \right) \frac{\boldsymbol{T}_{i}}{\|\boldsymbol{T}_{i}\|^{2}} - \frac{\omega \lambda}{2n} \frac{\nabla O(\boldsymbol{g}^{\lfloor k/n \rfloor - n})}{\|\boldsymbol{T}_{i}\|^{2}}, \qquad (1.5)$$

kde  $\omega$  je opět relaxační parametr, ačkoli jeho optimální hodnota může být odlišná od předchozí metody. Snadno se přesvědčíme, že pokud má tato iterace pevný bod, pak je jím řešení (1.3); Konvergence takto regularizované metody ovšem není zřejmá. Odvození podobného vzorce, avšak vycházejícího z metody SART, je k nalezení v [29].

Pro úplnost zmiňme, že existují regularizační metody, které "nevyužívají a-priori informace" [6]. Lze mezi ně zařadit iterační metody s omezeným počtem kroků, v jejichž případě se spoléhá na to, že vysokofrekvenční složky řešení v iteraci konvergují pomaleji než nízkofrekvenční. Sofistikovanějším algoritmem je *Truncated Singular Value Decomposition* (TSVD) [30]. Ten vychází z metody známé jako SVD (*Singular Value Decomposition*), jež rozkládá libovolnou matici na součin tří speciálních:

$$\mathbf{T} = \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^{\mathrm{T}}, \qquad (1.6)$$

kde matice **U** a **V** jsou ortogonální a  $\Sigma$  je diagonální (obecně obdélníková). Pokud je původní matice regulární, všechny diagonální prvky matice  $\Sigma$  (*singulární hodnoty*) jsou nenulové, a rozkladu lze využít k přímému nalezení řešení úlohy (inverzní matice k ortogonální matici se vypočte její transpozicí, u diagonální matice stačí invertovat jednotlivé diagonální prvky). U singulární matice se dá SVD využít k výpočtu její *pseudoinverze*, a to tak, že místo invertování všech diagonálních hodnot matice  $\Sigma$  invertujeme jen ty nenulové (a pokud není čtvercová, transponujeme ji).

Špatně podmíněná úloha se v SVD pozná podle toho, že některé singulární hodnoty jsou mnohem blíže nule než jiné. Dá se s ní zacházet tak, že singulární hodnoty menší než zvolený práh nahradíme nulami ("truncation") a místo inverze provedeme pseudoinverzi [30]. Tím z řešení odstraníme ty složky, které jsou hodně citlivé na odchylky v datech (což jsou u tomografie opět ty vysokofrekvenční). Výhodou tohoto přístupu je jeho výpočetní úspornost, protože singulární hodnoty jsou počítány v pořadí od největší k nejmenší, a jakmile je překročen práh, dá se výpočet ukončit. Za nevýhodu lze naopak považovat skutečnost, že TSVD neumí využít a-priori informace, pokud je nějaká k dispozici [6].

#### 1.5 Algoritmus na tokamaku COMPASS

Algoritmus, který se na tokamaku COMPASS používá k tomografii celkového vyzářeného výkonu a měkkého rentgenového záření, je založen na Tichonovově regularizaci s bází tvořenou obdélníkovými (případně čtvercovými) pixely a s předpokladem hladkosti profilu emisivity. Hladkost profilu lze vyjádřit různými způsoby, z nichž na COMPASSu se používají metody založené na minimalizaci normy některé prostorové derivace profilu, respektive minimalizace Fisherovy informace.

Minimalizace derivací využívá toho, že derivace fungují jako horní propust; Na funkci s významnou vysokofrekvenční složkou ve spektru nabývají vysokých hodnot, kdežto u pomalu se vyvíjejících funkcí jsou blízké nule. Rekonstrukční metoda regularizovaná jejich minimalizací tedy upřednostňuje hladké výsledky. Kromě toho má tento druh hodnotícího funkcionálu jednu algoritmickou výhodu. Původní Tichonovova regularizace počítá s tím, že hodnotící funkcionál je kvadratický a pozitivně definitní a dá se vyjádřit jako  $O(\mathbf{g}) = \| \Gamma \mathbf{g} \|^2$ (matice  $\Gamma$  se označuje jako Tichonovova); Minimalizace derivací tento tvar mají. Každý řádek jejich Tichonovovy matice vyjadřuje jistou konečnou diferenci, tedy lineární kombinaci dvou nebo více blízkých pixelů, jejíž hodnota aproximuje směrovou derivaci profilu v daném místě (přesná derivace funkce složené z pixelů je ovšem nepoužitelná, protože taková funkce má skoky). Zvláštní pozornost je přitom potřeba věnovat okrajovým podmínkám, protože i ty mají vliv na tvar řešení. Jednou z možností jejich implementace je zahrnout mezi konečné diference i absolutní hodnoty všech okrajových pixelů. To odpovídá podmínce pevných okrajů s nulovou hodnotou (amplitudy okrajových pixelů lze interpretovat jako rozdíl funkční hodnoty v pixelu a nulové hodnoty za jeho okrajem). Analogicky lze zavést volné okraje, totiž tak, že konečné diference na okrajích do hodnotícího funkcionálu nezařadíme. Regularizace používaná na COMPASSu obvykle počítá s pevnými okraji, případně s aritmetickým průměrem zmíněných dvou případů (v dalším textu označováno jako napůl pevné okraje).

Úloha regularizovaná funkcionálem v popsaném tvaru se dá vyřešit pomocí metod lineární algebry. Její minimalizační varianta (obdoba (1.3)) má tvar

$$\underset{g}{\operatorname{arg\,min}} \|\mathbf{T}\boldsymbol{g} - \boldsymbol{f}\|^{2} + \lambda \|\boldsymbol{\Gamma}\boldsymbol{g}\|^{2}.$$
(1.7)

Tato minimalizace se dá převést do podoby soustavy lineárních rovnic vyjadřujících, že derivace vztahu (1.7) podle kterékoli složky g musí být nulová:

$$(\mathbf{T}^{\mathrm{T}}\mathbf{T} + \lambda \mathbf{\Gamma}^{\mathrm{T}}\mathbf{\Gamma})\mathbf{g} - \mathbf{T}^{\mathrm{T}}\mathbf{f} = \mathbf{0}.$$
 (1.8)

Tuto soustavu pak vyřešíme některou z metod k tomu určených (například Gaussovou eliminací [23]). Vzhledem k výsledné podobě úlohy se tento tvar regularizace označuje jako *lineární*.

Fisherova informace je funkcionál zobrazující pravděpodobnostní rozdělení závislé na nějakém parametru na kladné reálné číslo; Toto číslo se dá interpretovat jako přesnost, s jakou lze odvodit hodnotu parametru z pozorování veličiny s daným pravděpodobnostním rozdělením [31]. Pro účely regularizace tomografické úlohy považujeme (normalizovaný) profil emisivity plazmatu za pravděpodobnostní rozdělení, podle kterého jsou emitovány fotony, a prostorové posunutí tohoto rozdělení (vzhledem k libovolnému referenčnímu bodu, na tom výsledek nezávisí) za parametr. Provádíme-li dvojrozměrnou tomografii, jsou tyto parametry dva; Můžeme vypočítat Fisherovu informaci pro každý z nich zvlášť a za hodnotu regularizačního funkcionálu dosadit jejich součet. Tak dostaneme tvar zavedený v [32]:

$$O(g) = \int_{\mathbb{R}^2} \frac{(\partial_x g)^2 + (\partial_y g)^2}{g} \, \mathrm{d}x \, \mathrm{d}y \tag{1.9}$$

kde g(x, y) je profil emisivity tvořený lineární kombinací bázových funkcí s koeficienty g. Je zřejmé, že pro numerické účely musíme vztah (1.9) diskrétně aproximovat. Podobně jako výše tedy nahradíme derivace konečnými diferencemi, které aproximují hodnotu derivace v určitém bodě, a do jmenovatele dosadíme hodnotu g v tom bodě. Na COMPASSu se používají dvě varianty tohoto přiblížení. V první se počítají rozdíly mezi dvěma pixely se společnou hranou a výsledná diference se považuje za přiblížení derivace ve středu jednoho z pixelů (tj. dopředná nebo zpětná diference). Za hodnotu g se pak dosadí přímo hodnota pixelu. Druhá varianta spočívá v interpretaci rozdílu mezi sousedními pixely jako jako centrální diference, která aproximuje derivaci funkce ve středu hrany mezi pixely. Řešení má v tom místě skok, ale dá se nahradit aritmetickým průměrem obou sousedících hodnot (což by mělo zřetelnější smysl při použití interpolativních pixelů, které ovšem využívány nejsou). V každém případě pak lze místo integrálního vztahu (1.9) psát

$$O(\boldsymbol{g}) = \sum_{i} \frac{(\Delta_{i} \boldsymbol{g})^{2}}{g(x_{i}, y_{i})} = \boldsymbol{g}^{\mathrm{T}} \boldsymbol{\Gamma}^{\mathrm{T}} \boldsymbol{W} \boldsymbol{\Gamma} \boldsymbol{g}.$$
(1.10)

V tomto zápisu matice  $\Gamma$  realizuje výpočet konečných diferencí prvního řádu  $\Delta_i$  (v obou směrech), a jako **W** jsme označili diagonální matici s prvky  $1/g(x_i, y_i)$ . Nyní je zřejmá podoba Fisherovy informace s kvadrátem normy prvních prostorových derivací; Liší se přítomností *váhové matice* **W**, která přiřazuje různý význam derivaci (a hladkosti) v různých místech profilu. V místech s vysokou emisivitou je na hladkost kladen nižší důraz, díky čemuž není emisivita v těchto oblastech uměle snižována. Naopak tam, kde je emisivita nízká, je dobře potlačen šum [32].

Na rozdíl od minimalizace normy derivací nemá Fisherova informace tvar lineární regularizace, a nelze na ni aplikovat postup vedoucí k soustavě lineárních rovnic, jako je (1.8); To proto, že matice **W** závisí na *g*. Lze ovšem použít aproximace navržené v článku [32]. Derivujeme minimalizační úlohu s hodnotícím funkcionálem (1.10) a zanedbáme přitom závislost **W** na *g*. Tím dostaneme soustavu nelineárních rovnic

$$(\mathbf{T}^{\mathrm{T}}\mathbf{T} + \lambda \mathbf{\Gamma}^{\mathrm{T}}\mathbf{W}\mathbf{\Gamma})\boldsymbol{g} - \mathbf{T}^{\mathrm{T}}\boldsymbol{f} = \boldsymbol{0}, \qquad (1.11)$$

21

kterou lze řešit metodou přímé iterace. Při ní opět zanedbáme závislost **W** na g a řešíme soustavu, jako by byla lineární. Získané řešení použijeme k aktualizaci **W**, a tento postup iterativně opakujeme. Princip přednostní minimalizace derivací v místech s nízkou emisivitou je tak zachován. Zjevnou nevýhodou tohoto přístupu je nutnost pro získání jedné rekonstrukce řešit více než jednu soustavu lineárních rovnic, byť se ukazuje, že jich zpravidla nemusí být mnoho [33].

Další výpočetní komplikace plyne z toho, že předem neznáme hodnotu parametru  $\lambda$ . Pro její určení se na COMPASSu využívá faktu, že hlavním úkolem regularizace je eliminovat vliv šumu přítomného v datech. Volíme tedy  $\lambda$  tak, aby reziduum neregularizované úlohy odpovídalo úrovni šumu, jakou v datech očekáváme. Toto kritérium je známé jako Morozovův princip [34], a lze jej popsat vztahem

$$\chi^{2}(\boldsymbol{g}(\lambda)) = \frac{1}{|f|} \sum_{i=1}^{|f|} \frac{(\boldsymbol{T}_{i} \boldsymbol{g}(\lambda) - f_{i})^{2}}{\sigma_{i}^{2}} = 1$$
(1.12)

kde  $\sigma_i$  představuje očekávaný šum v *i*-té složce dat. K výpočtu rezidua je bohužel nutné znát řešení úlohy s danou hodnotou  $\lambda$ . K řešení tohoto problému se přistupuje jako k numerickému hledání kořene obecné funkce jedné proměnné, kde vyčíslení funkce spočívá ve výpočtu regularizované tomografické rekonstrukce a následném dosazení řešení do (1.12). To vyžaduje iterativní postup, a pokud je rekonstrukce regularizována Fisherovou informací, pak dva iterativní postupy vnořené do sebe.

Bylo ukázáno, že iteraci pro minimalizaci Fisherovy informace lze provádět zároveň s hledáním optimální hodnoty  $\lambda$ , tj. v každém iteračním kroku aktualizovat  $\lambda$  i váhovou matici. K výraznému zvýšení efektivity výpočtu tím ovšem nedojde [35].

#### 1.6 Možnosti algoritmického urychlení

Z předchozího odstavce plyne, že pro získání jednoho tomografického snímku je třeba provést řadu iterací zahrnujících řešení soustavy lineárních rovnic o rozměru odpovídajícím požadovanému rozlišení snímku. V těchto iteracích se optimalizuje hodnota regularizačního parametru, a v případě Fisherovy informace také tvar regularizačního funkcionálu. Nabízí se několik způsobů, jak tento výpočet urychlit.

Zjevnou možností je snížení počtu pixelů a tím i velikosti soustavy lineárních rovnic. K řešení této soustavy se používají přímé metody, jejichž časová složitost asymptoticky roste s třetí mocninou počtu rovnic (a pixelů); Tento výpočet představuje nejnáročnější část každého iteračního kroku. Pro rozlišení tak velká, že lze použít asymptotické přiblížení, je tedy možné snížením rozlišení iterační krok značně urychlit. Není však dobré snižovat rozlišení příliš, protože se ukazuje, že od jisté velikosti pixelu výš začíná tvar řešení záviset na zvoleném rozlišení [6], což naznačuje, že dochází k nežádoucí regularizaci omezením prostoru řešení. Také je možné, že špatné výsledky při nízkém rozlišení souvisejí s diskretizačními chybami, které se zvětšují přinejmenším lineárně s šířkou pixelu, a potenciálně s chybami ve veličinách, které na této diskretizaci závisejí [33]. K o něco lepším výsledkům může vést nahrazení pixelů interpolativními funkcemi a souvisejícímu zpřesnění přiblížení spojitého profilu emisivity při dané velikosti báze, jak je zmíněno v [22]. Jednoduchou optimalizaci počtu pixelů lze také provést vynecháním těch pixelů, které leží mimo oblast, v níž se může vyskytovat plazma (například mimo komoru tokamaku). Drobným problémem tohoto přístupu je složitější tvar

okraje rekonstruované oblasti, a tím i okrajových podmínek.

Dalším způsobem, jak urychlit výpočet soustavy, je provést některé části výpočtu předem, a v kritické chvíli pak počítat jen to, co je nutné. Kdyby se matice soustavy mezi snímky (nebo alespoň mezi iteracemi jednoho snímku) neměnila, šlo by předpočítat matici k ní inverzní nebo její rozklad na součin matic, které se snadno invertují (například Chloleského faktorizaci nebo SVD). Po získání pravé strany by se pak provádělo pouze násobení maticí (v přímočaré implementaci kvadratická časová složitost) nebo řešení soustav s trojúhelníkovými maticemi (také kvadratické).

Matice tomografické úlohy se bohužel mezi snímky i mezi iteracemi mění, ovšem myšlenka počítání s předstihem zůstává užitečnou, použijeme-li obecnější metodu než ty zmíněné v minulém odstavci. Dobrou volbou je *Generalized Singular Value Decomposition* (GSVD), jak je popsáno například v [36]. Podobně jako u běžného SVD jde o rozkládání obecných matic na součiny matic diagonálních a ortogonálních; Tentokrát však rozkládáme najednou dvě matice se shodným počtem sloupců. Pro účely Tichonovovy regularizace lze GSVD zapsat takto:

kde matice U a V jsou ortogonální,  $\Sigma_1$  a  $\Sigma_2$  diagonální a Z regulární [36]. Lineárně regularizovanou úlohu (1.8) pak lze s pomocí tohoto rozkladu přepsat jako

$$(\mathbf{Z}^{\mathrm{T}}\boldsymbol{\Sigma}_{1}\mathbf{U}^{\mathrm{T}}\mathbf{U}\boldsymbol{\Sigma}_{1}\mathbf{Z} + \lambda \mathbf{Z}^{\mathrm{T}}\boldsymbol{\Sigma}_{2}\mathbf{V}^{\mathrm{T}}\mathbf{V}\boldsymbol{\Sigma}_{2}\mathbf{Z})\boldsymbol{g} = \mathbf{T}^{\mathrm{T}}\boldsymbol{f}, \qquad (1.14)$$

a následně, s využitím ortogonality matic U a V,

$$\mathbf{Z}^{\mathrm{T}}(\boldsymbol{\Sigma}_{1}^{2} + \lambda \boldsymbol{\Sigma}_{2}^{2})\mathbf{Z}\boldsymbol{g} = \mathbf{T}^{\mathrm{T}}\boldsymbol{f}. \qquad (1.15)$$

Nyní je možné rozložit úlohu na postupné řešení úloh s maticemi  $\mathbf{Z}^{T}$ ,  $(\boldsymbol{\Sigma}_{1}^{2} + \lambda \boldsymbol{\Sigma}_{2}^{2})$  a  $\mathbf{Z}$ ; První a poslední lze invertovat předem, prostřední je diagonální. Takto lze v reálném čase vyřešit lineárně regularizovanou úlohu s časovou složitostí kvadratickou v počtu rovnic, i když hodnota  $\lambda$  není předem známa. Problém nastává při použití Fisherovy informace, kdy se vztah (1.15) změní do tvaru

$$\mathbf{Z}^{\mathrm{T}}(\boldsymbol{\Sigma}_{1}^{2} + \lambda \boldsymbol{\Sigma}_{2} \mathbf{V}^{\mathrm{T}} \mathbf{W} \mathbf{V} \boldsymbol{\Sigma}_{2}) \mathbf{Z} \boldsymbol{g} = \mathbf{T}^{\mathrm{T}} \boldsymbol{f}.$$
(1.16)

Váhová matice **W**, která se přinejmenším v některých iteračních krocích mění, brání dalšímu zjednodušení úlohy; Matice uvnitř závorek navíc už obecně není diagonální. Lze alespoň uvažovat o využití GSVD k urychlení těch fází výpočtu, kdy se mění pouze  $\lambda$ , ovšem užitečnost takového postupu je nejasná, protože po každé změně váhové matice je třeba provést GSVD rozklad a inverzi tím získané matice **Z**.

Alternativním způsobem ke zjednodušování iteračního kroku je snížení potřebného počtu těchto kroků, a to pokud možno tak, aby neutrpěla přesnost výsledků. Je tedy potřeba urychlit konvergenci iterační metody. K určení vhodného  $\lambda$  z Morozovova principu lze použít některou ze standardních metod pro hledání kořene funkce, například metodu sečen, jejíž řád konvergence činí  $\varphi \approx 1.618$ . Rychlejší asymptotické konvergence lze dosáhnout s metodami založenými na interpolaci řešené funkce polynomem vyššího než prvního řádu, které mají tu výhodu, že nepotřebují znát žádnou derivaci funkce. Zmíněné metody jsou však použitelné jen

pro optimalizaci samotného parametru  $\lambda$ , protože jejich postup závisí na několika předchozích vypočtených hodnotách, a mezi výpočty těchto hodnot se nesmí změnit jiné parametry zkoumané funkce, jako například váhová matice. Pokud se váhová matice mění zároveň s  $\lambda$ , je vhodné použít stabilnější metodu, jako je metoda tětiv (neboli regula falsi) [35].

Použití metody s derivacemi, jako je Newtonova, je možné, ale vyžaduje to výpočty navíc. Derivujeme-li vztah pro výpočet  $\chi^2$  a předpokládáme-li přitom lineární regularizaci, dostaneme

$$\left(\frac{\partial \chi^2}{\partial \lambda}\right)_f = 2(\mathbf{T}\boldsymbol{g}(\boldsymbol{f},\lambda) - \boldsymbol{f})^{\mathrm{T}} \boldsymbol{\Sigma} \mathbf{T} (\mathbf{T}^{\mathrm{T}} \mathbf{T} + \lambda \mathbf{0})^{-1} \mathbf{O} \boldsymbol{g}(\boldsymbol{f},\lambda).$$
(1.17)

Pro Newtonovu metodu je tedy třeba řešit známou soustavu lineárních rovnic ještě jednou, s novou pravou stranou. To je naštěstí možné udělat tak, aby časová složitost přidaných výpočtů byla jen kvadratická v počtu rovnic. Na druhou stranu tím asymptoticky urychlíme hledání  $\lambda$ do řádu 2. Je ovšem třeba vzít v úvahu i obor konvergence Newtonovy metody a to, jak rychle konverguje, dokud je daleko od kořene (většinou nemáme zájem hledat  $\lambda$  s maximální numericky dosažitelnou přesností). Není tedy zřejmé, za jakých okolností se tento postup vyplatí.

K určování vah Fisherovy informace byl zatím používán jen algoritmus popsaný v minulém odstavci. Jeho implementace je jednoduchá, a konverguje s dostatečnou přesností během několika iterací. Každá z těch iterací ovšem vyžaduje řešení soustavy lineárních rovnic. Vedle přímočarého snižování složitosti jednotlivých kroků nebo jejich počtu se tedy nabízí zvolit metodu, která mezi oběma veličinami udělá rozumný kompromis. Může přitom jít i o metodu pro minimalizaci funkce, kterou aplikujeme na úlohu ve tvaru (1.3). Opět není snadné analyzovat, k jaké změně časové složitosti algoritmu tak dojde.

Konvergenci iteračních metod lze ještě urychlit vhodnou volbou počátečního přiblížení. Lze například vypočítat tomografickou rekonstrukci stejného snímku s nižším rozlišením a do požadovaného rozlišení ji interpolovat [37]. Jinou možností je předpokládat, že profil plazmatu se v čase vyvíjí poměrně pomalu (vzhledem k vzorkovací frekvenci tomografické diagnostiky), a po sobě následující snímky jsou tedy podobné. V tom případě můžeme jako počáteční přiblížení použít výsledek předchozí rekonstrukce. Podle práce [35] je takto možné zkrátit iteraci minimalizace Fisherovy informace na jeden krok (jednu soustavu lineárních rovnic na snímek), ovšem za cenu občasného výskytu nepřesně rekonstruovaných snímků.

S redukcí iterace na jeden krok se objevuje další využití pro rozklady matice soustavy: Rozklad lze provést dříve, než jsou načtena první data. S daty na pravé straně se pak řeší pouze soustavy s trojúhelníkovými, ortogonálními nebo diagonálními maticemi. Tím dosáhneme zkrácení času mezi vstupem dat a jim odpovídajícím výstupem tomografické rekonstrukce [38]. Vzorkovací frekvence však zůstává stejná, určená zejména tím, jak rychle umíme rozložit matici úlohy. S lehkým využitím paralelizace můžeme zvýšit i ji, přičemž se ještě sníží přesnost výstupů; Jednou rozloženou matici lze totiž používat k rekonstrukci dalších a dalších snímků, zatímco druhé vlákno programu připravuje rozklad nové matice. Je zřejmé, že matice, která je mírně neaktuální už tehdy, když je vypočten její rozklad, zastarává během svého dalšího používání dále. Záleží tedy na navazujících regulátorech, zda tento kompromis mezi rychlostí a přesností ocení.

Informace z předchozích rekonstrukcí se dá patrně použít i v případě kvadratických hodnotících funkcionálů. Tyto funkcionály nevyužívají derivace vážené převrácenými hodnotami neznámé, takže se z předchozích snímků může použít pouze odhad  $\lambda$ . Zajímavější situace v tomto (i původním) případě ovšem nastane, řešíme-li i soustavu lineárních rovnic některou z iteračních metod (ať už ART aplikovanou na (1.2) nebo obecnější metodou, kterou použijeme k řešení (1.8)), které podobně jako iterace Fisherovy informace vyžadují inicializaci počátečním odhadem řešení. Konvergence prakticky používaných metod bývá (alespoň pro určité typy soustav) zaručena s libovolnou inicializací, ovšem lze očekávat, že při použití dobrého počátečního odhadu bude k nalezení dostatečně přesného řešení opět stačit nižší počet iteračních kroků. S využitím informace z dříve rekonstruovaných snímků lze tedy nejen snížit počet iterací, ale i zkrátit výpočet jednoho jeho iteračního kroku.

Popisované problémy s nepřesnou rekonstrukcí, způsobenou omezením počtu iteračních kroků, se netýkají všech snímků, nýbrž jen těch, které následují po rychlé změně profilu emisivity, nebo které jsou poškozeny nějakou nestabilitou používaných iteračních metod. Byla navržena kritéria umožňující tyto poškozené snímky rozeznat [35]. Pokud je metoda použita mimo reálný čas, dají se rozpoznané snímky opravit například tak, že s jim odpovídajícími daty provedeme nějaké iterace navíc. V reálném čase to nejde, nicméně existence kritérií dovoluje případným regulátorům vadné snímky přeskočit nebo jim přikládat menší váhu, raději než na jejich základě provést chybný zásah. Je ovšem jasné, že i tak je třeba výskyt těchto chyb co nejvíce omezit.

## 2 Implementace tomografie

Tomografické a jim podobné metody nacházejí rozsáhlé uplatnění v analýze dat zaznamenaných během experimentu. S vývojem počítačů, respektive elektroniky navržené speciálně pro digitální zpracování signálu, se však pro ně objevuje odvážnější využití, totiž řízení experimentu v reálném čase. Odvážnější proto, že po výstupech algoritmů je kromě správnosti požadována i rychlost a spolehlivost.

Požadavky na rychlost typicky mají u softwaru pro reálný čas podobu *deadlinů*, nejvyšších dovolených zpoždění mezi vstupem a odpovídajícím výstupem programu. Úrovně požadované spolehlivosti se liší podle toho, jak závažné jsou následky zmeškání deadlinu: U některých aplikací výpočtů v reálném čase tím dojde k selhání řízeného systému, zatímco jinde je výsledek výpočtu užitečný i po uplynutí deadlinu, byť méně než při včasném dodání. Obecně však platí, že u systémů pro výpočty v reálném čase je kladen důraz na to, aby jejich výstupy přicházely v předvídatelných intervalech.

Pokud má být tomografie měkkého rentgenového záření použita například ke zpětnovazebnímu řízení vertikální polohy plazmatu, je nutné, aby její výpočet zabral významně kratší dobu, než kolik činí časová konstanta vertikální nestability. Ta je různá pro různé tokamaky, závisí na jejich velikosti (větší tokamaky mívají pomalejší vertikální nestabilitu) a na konstrukci komory (vířivé proudy ve vodivé stěně zpomalují pohyby plazmatu). U tokamaku COMPASS činí tato časová konstanta přibližně 0.5 ms [3], zatímco u tokamaku JET jsou to jednotky milisekund [39]. Předpokládá se, že pro tokamak ITER to budou desítky milisekund; Zde ovšem definice nestability závisí i na designu stabilizačního systému, který je schopný dodávat do řídicích cívek omezený proud a tím pádem má omezenou schopnost zachytit plazma, které se od rovnovážné polohy vzdálí více než na několik centimetrů [40]. U případného aktivního řízení transportu nečistot se situace zdá být příznivější, protože jev je pomalejší; Na tokamaku JET se jeho charakteristické trvání pohybuje ve stovkách milisekund [41]. Možnosti této technologie však dosud nejsou řádně prozkoumány a není tedy jasné, jaké ve výsledku budou požadavky na řídící algoritmus.

Tato kapitola se zabývá především optimalizací tomografického výpočtu, avšak je třeba mít na paměti, že to není jediná součást stabilizačního systému, na níž závisí úspěch experimentu. Tomografii předchází sběr dat a po ní následuje vyhodnocení výsledků. U řízení polohy jde o nalezení místa s maximální emisivitou, těžiště profilu nebo jiné podobné veličiny; Pro řízení transportu nečistot je třeba vyhodnotit, jakému rozložení nečistot získaný profil emisivity odpovídá. Výstup tohoto procesu pak představuje vstup pro regulátor, který ovšem obvykle nebývá moc výpočetně náročný (vertikální poloha plazmatu na tokamaku COMPASS je řízena PID regulátorem [3]). Dále je na řadě opět hardware, v tomto případě *aktuátory*, což jsou v případě vertikální stabilizace zdroje dodávající proud do stabilizačních cívek (na COMPASS se používají zdroje schopné měnit výstup jednou za 25 µs [3]) a cívky samotné (mají induktanci, a pokud jsou umístěny mimo vodivou komoru, je třeba počítat i s časem prostupu pole skrz tuto komoru). Celá zpětnovazební smyčka je znázorněna na obrázku 2.1. Výslednou veličinou, která určuje rychlost zpětné vazby, je přitom zpoždění mezi vstupem signálu do detektorů a nejbližším zásahem aktuátorů, který z tohoto signálu vychází.



Obr. 2.1: Schéma obecné smyčky zpětné vazby tokamaku (převzato z [42])

V současném řešení na tokamaku COMPASS jsou výpočty pro zpětnovazební řízení polohy prováděny s periodou 50 µs, tedy desetkrát kratší než časová konstanta nestability. Tato doba je ve zbytku tohoto textu považována za deadline pro tomografický výpočet polohy plazmatu. Je však třeba dodat, že průběh současného algoritmu, tedy zpoždění mezi vstupem a výstupem dat, trvá jen 12 µs. Jde o výpočet polohy z magnetických diagnostik, který je relativně jednoduchý, a existují pro něj efektivní algoritmy založené na vyhledávacích tabulkách [5]. Výpočet přitom provádí program napsaný v C++ běžící na běžném procesoru bez paralelizace.

Do následujících podkapitol jsou zařazeny výsledky z práce [43].

#### 2.1 Programovatelná hradlová pole

Obvyklé numerické výpočty nemají zcela sekvenční povahu, tj. není určeno jediné možné pořadí, v jakém je třeba vykonávat jejich dílčí operace. Pro urychlení těchto výpočtů se nabízí využít *paralelizace*, tedy provádět některé jejich části zároveň, na více výpočetních jádrech (procesorech nebo menších jednotkách). Podobně jako kombinace sekvenčního procesoru s pamětí umožňuje do jisté míry optimalizovat algoritmy volbou vhodného kompromisu mezi jejich časovou a prostorovou složitostí, u paralelních algoritmů lze navrhnout kompromis mezi třemi druhy složitosti: výpočetním časem, místem zabraným v paměti a počtem výpočetních jednotek. U výpočtů v reálném čase mají všechny tři veličiny svá omezení; Zatímco paměťové a procesorové nároky jsou omezeny dostupným hardwarem, na výpočetní čas jsou kladeny deadliny.

Samotná časová složitost u paralelních algoritmů navíc není popsána jen jedním číslem. Lze mluvit o veličinách zvaných *zpoždění* (*latence*) a *propustnost* (*throughput*), z nichž první označuje čas, který uplyne mezi vstupem dat a jim odpovídajícím výstupem, zatímco druhá

popisuje frekvenci, s jakou je zařízení schopno přijímat nová data (u výstupů se předpokládá, že jsou generovány se stejnou frekvencí). U obvodů s nastavitelným taktem se zavádí ještě třetí časová veličina zvaná *timing*; Ta říká, při jakém nejvyšším taktu obvod funguje správně [44]. Zpoždění a propustnost je opět možné optimalizovat do jisté míry zvlášť, zejména pomocí mechanizmu známého jako *pipelining*. Ten je založen na stejné myšlence jako montážní linka: Výpočetní proces je rozdělen na řadu navazujících úkonů, a každý je prováděn na jiném výpočetním jádru. Dosažitelná frekvence vstupů dat pak odpovídá frekvenci, s jakou data procházejí nejpomalejším jádrem, což ovšem bývá rychlejší než kdyby každé jádro provádělo celý výpočet [44]. Zpoždění zůstává stejné nebo se mírně zhorší, protože mezi vstupem a odpovídajícím výstupem musí postupně proběhnout dílčí výpočty na všech jádrech, plus komunikace mezi jádry. Přitom právě zpoždění je klíčovou veličinou u výpočtů v reálném čase.

Pro implementaci paralelních algoritmů byla vyvinuta řada typů hardwaru, od vícejádrových procesorů přes grafické karty po integrované obvody s konfigurovatelnou strukturou. Ne všechny se ovšem hodí k tomografii plazmatu v reálném čase. Komerčně dostupné vícejádrové procesory mají obvykle jader poměrně málo, čímž sice dovolují optimalizaci pomocí vícevláknového běhu programu, ale ne masivní paralelizaci některých fází tomografického výpočtu, jako je generování matice regularizované úlohy nebo následné řešení soustavy lineárních rovnic. Grafické karty nabízejí zajímavější možnosti paralelizace, výpočetních jader mívají hodně, a tato jádra jsou vysoce optimalizovaná pro numerické výpočty. Mimo jiné umožňují provést jeden druh operace na mnoha různých balících dat zároveň, díky čemuž jsou implementace metod pro řešení lineárních soustav na grafických kartách velmi rychlé. Problémem se však ukazuje být rychlost načítání dat, která je u grafických karet poměrně nízká; Vzhledem k tomu, že přenos dat je zapotřebí na začátku a na konci každého výpočtu, jeho pomalost poněkud zhoršuje možnosti použití grafických karet v reálném čase (alespoň v tak reálném, jaký třeba u zpětné vazby na tokamaku) [45].

Naopak jako vhodné pro zamýšlený druh výpočtů se ukazují integrované obvody známé pod názvem *programovatelná hradlová pole* (anglicky *Field-Programmable Gate Array*, FPGA). Ty nemají podobu souboru výpočetních jader určených pro paralelní počítání, nýbrž pravidelné mřížky relativně nízkoúrovňových součástek doplněných o infrastrukturu umožňující volitelné propojení a konfiguraci těchto součástek [44]. Taková struktura umožňuje s pomocí univerzálního zařízení dosáhnout podobných výsledků jako s použitím integrovaných obvodů vyráběných na míru dané aplikaci, čímž odpadá fáze vývoje a testování hardwaru ("Field-Programmable" lze chápat jako protiklad k "Factory-programmable"). Oproti integrovaným obvodům na míru je ovšem efektivita takovýchto čipů nižší pokud jde o rychlost, složitost designu realizovaného čipem dané velikosti i energetickou náročnost [46]. To je dáno přítomností konfigurační infrastruktury a faktem, že jednotlivé součástky jsou potenciálně mnohoúčelové. Obvody FPGA jsou tak vhodné spíše pro aplikace malého rozsahu, pro které se nevyplatí vyvíjet obvod na míru.

Univerzální struktura FPGA se skládá z *logických bloků*, které jsou obvykle složeny z konfigurovatelných vyhledávacích tabulek a jednobitových paměťových jednotek (*flip-flopů*). Tyto elementy mohou sloužit jako kombinatorická logika, paměť typu ROM, případně RAM (*distribuovaná paměť*) [47]. Kromě této univerzální struktury lze v moderních obvodech FPGA

nalézt nízkoúrovňové implementace často používaných prvků, jako jsou sčítačky a násobičky pro pevnou i plovoucí [48] řádovou čárku nebo paměťové bloky s větší kapacitou, než jakou dovoluje distribuovaná paměť (zvané *bloková paměť*).

Součástky v FPGA lze přímo spojovat do celků, které realizují požadované zobrazení ze vstupů na výstupy. Takovému designu se říká kombinatorický, a je vhodný spíše pro jednodušší funkční celky; Přítomnost složitých kombinatorických obvodů v designu totiž kazí timing. Obvody vykonávající složitější výpočty se navrhují s využitím registrů, které od sebe oddělují kombinatorické části; Registry pouští výstup jedněch obvodů na vstup druhých na základě hodinového signálu, který je obvykle společný pro celé FPGA a distribuovaný po síti, která zaručuje jeho synchronnost ve všech koutech obvodu [49]. Díky tomu lze v FPGA implementovat sekvenční obvody, a vyřešit tak paralelizaci algoritmu například implementací řady propojených procesorových jader, která mohou vykonávat běžný strojový kód. Takový přístup však není příliš efektivní ve srovnání s vícejádrovými procesory vyráběnými na míru svému účelu. Lepší je tvořit design pro FPGA jako kombinaci malých a jednoúčelových obvodů, jejichž spolupráce se dá ladit s přesností na jednotlivé takty hodinového signálu [44]. Podobně se dá optimalizovat vstup a výstup dat, což představuje značnou výhodu FPGA oproti grafickým kartám. Z hlediska využití čipu je dobré, aby optimalizovaná práce dílčích obvodů tvořila většinu spotřebovaného výpočetního času; Toho však lze často dosáhnout jen vhodnou volbou algoritmu, kterým budeme úlohu řešit.

Pro vývoj obvodů realizovaných v FPGA (i pro vývoj integrovaných obvodů na míru) se používají jazyky označované jako *Hardware Description Language* (HDL), z nichž nejrozšířenější jsou VHDL [50] a Verilog [51]. Tyto jazyky umožňují popsat digitální elektrický obvod podobně jako kreslené schéma. Na rozdíl od schématu ovšem dovolují i relativně vysokoúrovňové konstrukce, jako je opakování věcí ve smyčce, návrh konečných automatů nebo práce s datovými typy. Pomocí specializovaného softwaru lze HDL kód převést (*syntetizovat*) do podoby konfiguračního souboru pro FPGA. Existují ovšem i nástroje kompilující HDL do strojového kódu, který se dá spustit na sekvenčním procesoru (paralelní procesy jsou ovšem pouze simulovány). To dovoluje testovat fungování designu ještě před samotným naprogramováním součástky [52].

Díky jazykům HDL se návrh digitálních obvodů do jisté míry podobá běžnému programování. Stále je ovšem třeba mít na paměti některé odlišnosti tohoto procesu. Zejména je třeba počítat s tím, že návrh nemůže být libovolně složitý; Zatímco na sekvenčním hardwaru to vede pouze k vyšší spotřebě paměti nebo procesorového času, do FPGA se příliš složitý design nevejde. U aplikací pro rychlé zpracování dat je také třeba brát ohled na to, jak dlouho trvá, než signál projde jednotlivými částmi obvodu, případně spojem mezi nimi. Proto nástroje pro syntézu HDL většinou umožňují doplnit kód o podmínky kladené na timing.

#### 2.2 Řešení soustav lineárních rovnic

Soustavy lineárních rovnic se vyskytují v procesu řešení rozličných úloh, kromě tomografie plazmatu zejména při numerickém řešení diferenciálních rovnic objevujících se v řadě fyzikálních problémů, ale také například ve zpracování obrazu [53] nebo kryptoanalýze [54]. V řadě případů mají tyto soustavy zvláštní tvar, jehož lze při jejich řešení využít. Za tím účelem byla vyvinuta řada specializovaných algoritmů, z nichž některé se ukazují být vhodné i pro řešení tomografické úlohy. Navíc často využívají jednoduchých operací prováděných s mnoha prvky matice, což poskytuje prostor pro paralelizaci.

V dosavadní implementaci tomografického algoritmu se řeší soustavy s maticemi ve tvaru  $\mathbf{T}^{T}\mathbf{T} + \lambda \mathbf{0}$ , kde  $\mathbf{T}$  je projekční matice a  $\mathbf{0}$  matice (linearizovaného) hodnotícího funkcionálu. Matice  $\mathbf{0}$  je symetrická a obvykle pozitivně definitní (jde například o druhou mocninu matice konečných diferencí). Součin  $\mathbf{T}^{T}\mathbf{T}$  je symetrický a pozitivně semidefinitní, ať má matice  $\mathbf{T}$  jakýkoli tvar. Parametr  $\lambda$  se navíc volí kladný. Ve výsledku je symetrická a pozitivně definitní celá matice úlohy. Díky těmto vlastnostem lze namísto univerzálních metod, jako je Gaussova eliminace, použít například Choleského faktorizaci, jejíž výpočet je rychlejší. Z hlediska asymptotické složitosti jsou ovšem obě zmíněné metody srovnatelné; V sekvenční variantě mají časovou složitost  $\mathbf{0}(n^{3})$ , kde n je počet rovnic.

Matice **T** je (u pixelové tomografie) řídká, protože detektory jsou uspořádané tak, aby každý zabíral jen malou část plazmatu (tj. malý počet pixelů); Matice **O** je také řídká, například pro Fisherovu informaci dvojrozměrného profilu má v každém řádku nanejvýš pět nenulových čísel. Ačkoli pro pravidelně uspořádané pixely mívá matice hodnotícího funkcionálu několikadiagonální strukturu, celá matice úlohy takovou strukturu nemá; Nelze tedy použít vysoce efektivní metody určené pro tento případ. Řadu algoritmů lze ovšem přizpůsobit, aby využívaly alespoň řídkosti matice. V některých případech jde jen o přeskakování výpočtů s prvky, o nichž je známo, že jsou nulové (a o změnu reprezentace matice, která s tím souvisí). V jiných případech dojde ke značnému zesložitění algoritmu (například analyzováním struktury řídké matice přes grafové algoritmy [55]), i když výpočetní čas na sekvenčním procesoru i množství zabrané paměti se tím sníží. Z hlediska FPGA může být zesložitění algoritmu nežádoucí, protože vede ke zvýšení množství potřebných logických prvků.

Podobně jako metody pro tomografickou inverzi se i metody pro řešení soustav lineárních rovnic dají rozdělit na přímé a iterativní. Přímé metody řeší soustavu lineárních rovnic v jednom kroku, alespoň v tom smyslu, že použitelné řešení se objeví až po dokončení všech výpočtů; Toto řešení je až na chyby způsobené výpočty s digitálně reprezentovanými reálnými čísly přesné. Výše zmíněná Gaussova eliminace a Choleského faktorizace jsou příklady přímých metod. Iterační algoritmy oproti tomu pracují v mnoha krocích, přičemž po každém z těchto kroků je k dispozici nová aproximace řešení. Čím více kroků vykonáme, tím více se výsledek blíží správnému řešení – pokud algoritmus *konverguje*. Existují také algoritmy, které dokážou poskytnout numericky přesný výsledek v konečném čase, ale produkují i mezivýsledky, které se postupně blíží přesnému řešení podobně jako u iteračních metod; Příkladem takového algoritmu je metoda konjugovaných gradientů [56].

Pro praktické numerické výpočty mají iterační algoritmy několik výhod. Za prvé, jejich dílčí krok bývá výpočetně jednodušší než velký krok přímých metod, a lépe se optimalizuje pro rozsáhlé soustavy nebo omezený hardware. Často spočívá v násobení mezivýsledku maticí nebo řešení soustavy s trojúhelníkovou částí původní matice, což je dobré v tom, že hodnoty v matici není třeba přepisovat. Fakt, že se matice nemění, usnadňuje případnou paralelizaci, protože jednotlivé kroky výpočtu nemusejí čekat, až pro ně bude matice připravena z předchozích kroků (byť kroky mohou být sekvenčně závislé i jinými mechanismy). Také to pomáhá při optimalizaci algoritmů pro řídké matice, protože neměnnost matice zaručuje, že se neobjeví nové nenulové prvky, a reprezentaci matice lze snadněji přizpůsobit schématu, v

jakém probíhá přístup k jejím prvkům.

Dalším rysem iteračních metod, který lze považovat za výhodu, je možnost přerušit výpočet po zvoleném počtu iteračních kroků a dosáhnout tak kompromisu mezi krátkým výpočetním časem a vysokou přesností výsledku. Pro výpočty v reálném čase to znamená, že lze v rámci deadlinu získat alespoň nějaký výsledek, i když kompletní výpočet přímé metody by se nestihl. U některých iteračních metod se navíc ukazuje, že mezivýsledky konvergují k přesnému řešení v různých ohledech různě rychle [55]. U úloh podobných tomografii to může znamenat, že nízkofrekvenční složky řešení konvergují jinak rychle než vysokofrekvenční. Použijeme-li metodu s pomalou konvergencí vysokofrekvenčních složek k řešení regularizované úlohy, lze iteraci přerušit poté, co nízkofrekvenční složky skonvergují (za předpokladu, že vysokofrekvenční složky nebyly v přiblížení řešení přítomny od začátku a není třeba čekat, až zmizí).

Ještě jednou zajímavou vlastností iteračních metod je to, že vycházejí z počátečního přiblížení výsledku. Ačkoli počáteční přiblížení obvykle nemá vliv na asymptotickou rychlost konvergence metody [57], může mít vliv na počet kroků nutných k získání výsledku s praktickou přesností. Toho lze využít, pokud máme o řešení přibližnou představu už před začátkem výpočtu. Jak je zmíněno v odstavci 1.6, takovou představou může být řešení získané v minulém časovém kroku. Jinou možností je použít jako počáteční přiblížení interpolovaný výsledek získány při nižším rozlišení, podobně jako [37] navrhuje pro urychlení iterativní minimalizace Fisherovy informace. Lze dokonce uvažovat o kombinaci těchto dvou přístupů a o získání tomografické obdoby multigridních metod, používaných při řešení diferenciálních rovnic [58].

#### 2.3 Gaussova eliminace

Gaussova eliminace je přímou metodou použitelnou pro libovolný typ soustavy lineárních rovnic. Soustava definovaná maticí a pravou stranou se v ní transformuje na soustavu s jinou maticí i pravou stranou, avšak se shodnou množinou řešení. Nejznámější varianta Gaussovy eliminace spočívá v převedení soustavy do trojúhelníkového tvaru a jejího následného vyřešení pomocí *zpětného dosazování* [23]. Při zpětném dosazování se využije toho, že v soustavě s (horní) trojúhelníkovou maticí se vyskytuje rovnice závislá jen na posledním prvku neznámé, která se dá snadno vyřešit; Po jejím vyřešení se hodnota poslední neznámé dosadí do předchozí rovnice, v níž tím opět zůstane jen jedna neznámá (předposlední). Takto lze postupně vyřešit všechny rovnice. Jinou možností pro Gaussovu eliminaci je převést matici soustavy do diagonálního tvaru, takže pak zpětné dosazování není nutné. Tento postup bývá označován jako Gaussova – Jordanova eliminace [59].

Změna tvaru soustavy je realizována *řádkovými úpravami*, z nichž každá mění jeden řádek matice a příslušný prvek pravé strany. V nejjednodušší situaci stačí k eliminaci jeden druh řádkové úpravy, kterým je přičtení násobku řádku k některému z následujících řádků. K trojúhelníkovému tvaru matice to vede díky tomu, že násobky přičítaných řádků jsou voleny tak, aby úpravou ubyl jeden nenulový prvek pod diagonálou matice. Tento postup ovšem funguje jen pro matice *silně regulární*, tedy takové, u nichž je regulární i každá submatice vzniklá výběrem prvních *k* řádků a *k* sloupců. U matic, které nejsou silně regulární, může nastat stav, v němž některý nenulový poddiagonální prvek nelze eliminovat, protože řádek,

jehož násobek by bylo třeba odečíst, má v kritickém místě nulu. V takovém případě si lze pomoci zavedením nové řádkové úpravy, kterou je záměna pořadí řádků.

S Gaussovou eliminací souvisí na první pohled odlišná metoda zvaná LU faktorizace. V ní se neupravuje soustava včetně pravé strany, ale jen matice, a výsledkem je matice vyjádřená jako součin dvou trojúhelníkových matic (dolní a horní, proto "LU"). Následně se pomocí zpětného dosazování vyřeší dvě vnořené soustavy s těmito trojúhelníkovými maticemi. Souvislost s Gaussovou eliminací začne být patrná, vyjádříme-li řádkové úpravy jako násobení jednoduchými maticemi. Úprava "přičtení násobku řádku k nižšímu řádku" je volena tak, že matice jí odpovídající je dolní trojúhelníková; Celou Gaussovu eliminaci tak lze popsat jako násobení matice soustavy (a pravé strany) dolní trojúhelníkovou maticí. Označíme-li výslednou eliminovanou matici (podle popsaného algoritmu horní trojúhelníkovou) jako **U**, pak **L** je matice inverzní k matici realizující řádkové úpravy; Matice inverzní k dolní trojúhelníkové je přitom také dolní trojúhelníková. Jestliže LU faktorizace převádí soustavu do tvaru

$$\mathbf{L} \mathbf{U} \mathbf{x} = \mathbf{b}, \tag{2.1}$$

pak cílem Gaussovy eliminace je tvar

$$\mathbf{U}\,\mathbf{x} = \mathbf{L}^{-1}\,\boldsymbol{b}\,,\tag{2.2}$$

kde L a U označují tytéž matice.

Tvar matice **L** by se ovšem pokazil při použití úpravy typu "záměna řádků", jejíž matice má nenulový prvek nad diagonálou. Odtud je zřejmé, že popsaný postup LU faktorizace nelze vykonat pro libovolnou matici, ovšem pro silně regulární matice to jde. Vypočítat LU faktorizaci matice je podobně složité jako provést Gaussovu eliminaci (ve skutečnosti to lze udělat pomocí mírně modifikovaného algoritmu Gaussovy eliminace [60]). Její výhodou je ovšem nezávislost hlavního výpočtu na pravé straně, díky čemuž ho lze provést dříve, než je pravá strana známa, respektive provést ho jednou a výsledný rozklad pak používat k řešení soustavy s různými pravými stranami (které se mohou objevovat postupně).

Bohužel se ukazuje, že Gaussova eliminace i LU faktorizace ve výše popsané podobě trpí *numerickou nestabilitou.* Při té dochází ke zvýrazňování chyb způsobených například výpočty v omezené digitální reprezentaci čísel, a to i tehdy, když je řešená úloha dobře podmíněná [61]. Zejména jde o to, že při postupném sčítání násobků řádků občas vznikají prvky s malou absolutní hodnotou a velkou relativní chybou. V některém z dalších kroků může být takový prvek zvolen za *pivot*, tedy za číslo, jehož převrácenou hodnotou se násobí řádek určený k přičtení k jinému řádku. Touto operací dojde ke zvětšení absolutní chyby přítomné v koeficientu a jejím rozšíření na celý řádek; Kvůli tomu je následující postup eliminace ještě více nepřesný. U velkých soustav můžou tímto mechanismem narůst chyby natolik, že v řešení dominují [61].

Přesto je Gaussova eliminace použitelná, protože numerické nestabilitě se dá vyhnout zavedením mechanismu nazývaného *pivotace*. Při pivotaci během eliminace zaměňujeme pořadí řádků, případně i sloupců, a to tak, aby na místě pivotu v každém kroku stál co největší prvek (tj. prvek, u kterého je naděje na malou relativní chybu). Podobně lze upravit i LU

faktorizaci, pokud výsledný rozklad doplníme o permutační matice popisující, jak byly zaměněny řádky a sloupce [60]. Je řada možností, jak pivotaci provést, přičemž tyto možnosti jsou různě výpočetně složité, různě účinné a také různě vhodné pro paralelizaci [62]. Nejobvyklejšími možnostmi jsou *úplná* a *částečná pivotace*, přičemž v úplné se vybírá pivot z celého bloku matice, který je ještě třeba eliminovat (zaměňují se řádky i sloupce), zatímco u částečné se vybírá jen z prvků v daném sloupci. Úplná pivotace zaručuje dobrý výsledek za všech situací, ale má kubickou časovou složitost, která se projeví na složitosti celého algoritmu. Rovněž negativně ovlivňuje možnosti paralelizace [62].

Bez použití pivotace dovoluje Gaussova eliminace rozsáhlou paralelizaci. Operace přičtení násobku řádku k jinému řádku se skládá z řady navzájem nezávislých výpočtů, jedinou komplikací je přitom potřeba distribuce (*broadcastu*) koeficientu mezi výpočetní jádra. Další paralelizace se nabízí v tom, že je možné provést přičtení (různých) násobků jednoho řádku k více řádkům zároveň. Tentokrát je přitom potřeba kromě broadcastu koeficientů provést také broadcast prvků řádku, protože každý prvek je používán ve více výpočetních jádrech. Za určitých podmínek lze provést broadcast v konstantním čase; Dovoluje to například architektura FPGA využívající *sběrnic*, které jsou jedním výpočetním jádrem nastaveny na zvolenou hodnotu a všechna ostatní jádra zároveň tuto hodnotu přečtou. V takovém případě lze řádkovou paralelizací dosáhnout kvadratické časové složitosti s lineárně rostoucím počtem výpočetních jader. U úplné blokové paralelizace je časová složitost lineární při kvadratické spotřebě výpočetních jader. Paměťová složitost je vždy kvadratická, protože je třeba uchovávat matici o velikosti  $n \times n$ .

Zmíněných časový složitostí je ovšem možné dosáhnout i v případě, že neumíme provést broadcast v jednom kroku; Lze jej totiž nahradit pipeliningem, tedy místo rozeslání informace všem výpočetním jednotkám najednou ji poslat jen první jednotce v řadě a nechat ji, aby ji v příštím kroku předala dál, zatímco sama přijme novou [63]. Konstantní koeficient u časové složitosti je pak ovšem horší než při použití sběrnic, což souvisí s tím, že na začátku a na konci výpočtu je aktivní jen část výpočetních jader (na začátku ta, ke kterým už se dostala rozesílaná informace, a na konci ta jádra, ke kterým se dostala až teď).

Gaussova eliminace ve své diagonalizační variantě je prvním algoritmem vybraným pro implementaci na FPGA v rámci této práce. Tato varianta je v sekvenční podobě pomalejší než eliminace do trojúhelníkového tvaru následovaná zpětným dosazením, protože tráví víc času ve fázi výpočtu s kubickou časovou složitostí. Ve zvolených úplně řádkově, respektive blokově paralelizovaných implementacích je ovšem naopak rychlejší, a pro implementaci na FPGA má tu výhodu, že nepotřebuje zvláštní obvody určené pro zpětné dosazování. Řádkově paralelizovaná implementace v každém případě potřebuje *n* výpočetních jader, blokově paralelizovaná  $n^2$ ; Ke zpomalení nedojde proto, že operace navíc provedou výpočetní jádra, která by jinak nebyla využita (zpracovávaná část řádků se postupně zkracuje, jak od začátku řádků přibývají nulové prvky). Obě uvažované implementace metody ukazuje kód 2.1.

```
1 void gaussJordan_radkovy(double** A, double* b, double* x, int n){
 2
       // A .. matice, b .. pravá strana, x .. předchozí aproximace řešení,
 3
       // x2 .. místo na novou aproximaci, n .. počet neznámých
 4
       // inicializace pivotace
 5
 6
       int maxPivot = 0;
 7
       for(int i=1; i<n; i++){</pre>
 8
            if(abs(A[i][0]) > abs(A[maxPivot][0])){
 9
                maxPivot = i;
10
           }
11
       }
12
       // záměna řádků
13
       double* radek = A[0]; A[0] = A[maxPivot]; A[maxPivot] = radek;
       double prvek = b[0]; b[0] = b[maxPivot]; b[maxPivot] = prvek;
14
15
       // eliminace s pivotací
16
       for(int i=0; i<n; i++){</pre>
17
           maxPivot = i+1;
18
           for(int j=0; j<n; j++){</pre>
19
                if(i==j)continue;
20
                double koef = A[j][i]/A[i][i];
21
                parfor(int k=i+1; k<n; k++)A[j][k] -= koef*A[i][k];</pre>
22
                b[j] -= koef*b[i];
23
                if(j>i && abs(A[j][i+1]) > abs(A[maxPivot][i+1])){
24
                    maxPivot = j;
25
                }
26
           }
           if(maxPivot!=i+1){
27
28
                radek = A[i+1]; A[i+1] = A[maxPivot]; A[maxPivot] = radek;
29
                prvek = b[i+1]; b[i+1] = b[maxPivot]; b[maxPivot] = prvek;
30
           }
31
       }
       // rekonstrukce řešení
32
33
       parfor(int i=0; i<n; i++){</pre>
34
           x[i] = b[i]/A[i][i];
35
       }
36 }
37
38 void gaussJordan_blokovy(double** A, double* b, double* x, int n){
39
       // eliminace bez pivotace
40
       for(int i=0; i<n; i++){</pre>
41
            parfor(int j=0; j<n; j++){</pre>
42
                if(i==j)continue;
43
                double koef = A[j][i]/A[i][i];
44
                parfor(int k=i+1; k<n; k++)A[j][k] -= koef*A[i][k];</pre>
45
                b[j] -= koef*b[i];
           }
46
47
       }
       // rekonstrukce řešení
48
49
       parfor(int i=0; i<n; i++){</pre>
50
           x[i] = b[i]/A[i][i];
51
       }
52 }
```

Kód 2.1: Řádkově (nahoře) a blokově paralelizovaná Gaussova – Jordanova eliminace. Klíčovým slovem **parfor** jsou označeny cykly vhodné k paralelnímu vykonání. V implementaci řádkově paralelizované eliminace se počítá s řádkovou pivotací, blokově paralelizovaná varianta je implementována bez pivotace. V obou případech je zanedbána implementace režie

související s výpočtem na více jádrech.

#### 2.4 Choleského faktorizace

Je-li matice soustavy symetrická a pozitivně definitní, lze provést LU faktorizaci tak, že  $\mathbf{U} = \mathbf{L}^{\mathrm{T}}$ . Tento druh rozkladu se nazývá Choleského faktorizace, a oproti obecné LU faktorizaci má několik výhod. Z optimalizačního hlediska je dobré, že ho lze provést přibližně dvakrát rychleji než LU faktorizaci nebo Gaussovu eliminaci, což souvisí s tím, že počítaných prvků je dvakrát méně (asymptotická časová složitost v konvenční implementaci ovšem zůstává kubická) [64]. Rovněž je dobré, že za předpokladu symetrické a pozitivně definitní matice lze Choleského faktorizaci provést vždy, aniž by bylo třeba zaměňovat řádky a sloupce matice (submatice tvořené prvními *k* řádky a sloupci pozitivně definitní matice jsou vždy regulární). Navíc je výpočet Choleského faktorizace numericky stabilní [64].

Choleského faktorizaci lze upravit pro rozklad matice na součin typu  $LDL^{T}$ , kde L je dolní trojúhelníková a D je diagonální; Pro jednoznačnost lze předpokládat, že L má na diagonále samé jedničky. Tento rozklad je mimo jiné možné provést i pro některé matice, které nejsou pozitivně definitní, ačkoli pak není zaručena jeho numerická stabilita [65]. Z hlediska případné implementace na FPGA je však jeho zásadní výhodou skutečnost, že není potřeba počítat žádné odmocniny.

Nevýhodou Choleského faktorizace je vyšší sekvenční závislost výpočtů, než jakou má Gaussova eliminace. To je špatné nejen pro paralelizaci, ale například i z hlediska přístupu do vyrovnávací paměti u sekvenčního procesoru. Závislost dat lze do jisté míry omezit vhodným přerovnáním matice, a také použitím **LDL**<sup>T</sup> rozkladu [66]. Konvenční implementace Choleského faktorizace pracuje se třemi vnořenými cykly, jejichž hierarchii lze měnit; Spolu s ní se mění i schéma přístupu k datům. Samotný tento postup nevede k (významné) optimalizaci, ale lze jej doplnit o volbu reprezentace matice (např. po řádcích), která se schématem přístupu dobře spolupracuje. Dalšího vylepšení je možné dosáhnout použitím *blokových algoritmů*, které rozdělí matici na menší bloky a provádějí jejich faktorizaci [65]. Po většinu výpočetního času je tak přístup k datům omezen na obsah jednotlivých bloků. Algoritmus se však zjevně stává složitějším.

Na algoritmu Choleského faktorizace lze předvést, jaké možnosti optimalizace poskytuje řídkost matic. První možností je vynechat všechna přičítání násobků nulových prvků, dělení nulových prvků koeficienty a počítání jejich odmocnin. Lze snadno ukázat, že tím dojdeme k algoritmu, který provádí aritmetické operace v počtu úměrném součtu kvadrátů počtů nenulových prvků ve sloupcích výsledné matice L [67], zatímco u algoritmu nepočítajícího s řídkostí matice je to úměrné součtu druhých mocnin počtů všech prvků ve sloupcích L. Patrně tedy nejde o zhoršení, ovšem není zcela snadné analyzovat, o jaké jde zlepšení. Struktura nenulových prvků matice L totiž nemusí odpovídat původní matici.

Během výpočtu Choleského faktorizace obecně dochází v matici k *zaplňování*, tedy objevují se nové nenulové prvky. Rozsah zaplňování závisí na struktuře původní matice, přičemž k němu nemusí dojít vůbec, ale také může být matice zaplněna úplně. Pozice nových nenulových prvků lze odvodit s využitím algoritmů známých jako *symbolická eliminace*; Tuto analýzu lze provést v čase lineárně rostoucím s počtem nenulových prvků v matici L [67]. Symbolická eliminace je užitečná k optimalizaci paměťových nároků algoritmu (poskytuje informaci o tom, kolik místa je třeba vyhradit pro nové prvky), ale i pro teoretickou analýzu způsobů, jak zaplňování

omezit.

Omezení zaplnění lze často dosáhnout záměnou pořadí řádků a sloupců v matici (obojí zároveň, aby matice zůstala symetrická). Například se ukazuje, že je dobré převést matici do co nejužšího pásového tvaru; K zaplnění pak může docházet pouze uvnitř pásu. Nalezení optimálního přerovnání matice (ne nutně pásového) je bohužel NP-úplná úloha [65], ovšem i prakticky použitelnými heuristickými postupy lze dosáhnout významné úspory paměti a zrychlení výpočtu. Někdy lze navíc využít toho, že má být faktorizováno více matic se stejnou strukturou (což je případ tomografické úlohy regularizované minimalizací Fisherovy informace) a provést analýzu a návrh přerovnání pouze jednou. Implementace v FPGA pak může počítat s konkrétním tvarem matice a nemusí zahrnovat algoritmus pro symbolickou faktorizaci.

#### 2.5 Jacobiho a Gaussova – Seidelova metoda

Iterační metody pro řešení lineárních soustav se dají rozdělit do dvou skupin, někdy nazývaných *stacionární* a *nestacionární* [57]. Nestacionární typicky konvergují rychleji, ale jsou algoritmicky složitější; Vzhledem k tomu, že na hardwaru typu FPGA je algoritmická složitost kritická, byla pro implementaci vybrána jedna ze stacionárních metod, konkrétně metoda Gaussova – Seidelova.

Gaussova – Seidelova a jí podobná Jacobiho metoda spočívají v rozdělení matice soustavy na součet dvou částečně zaplněných matic. Následně lze soustavu přepsat do ekvivalentního tvaru, v němž se neznámý vektor vyskytuje na dvou místech, z toho jednou explicitně. Tento tvar použijeme jako iterační předpis, kde na místo implicitního výskytu neznámé dosadíme předchozí iteraci a místo toho explicitního novou iteraci; Tu pak ze vztahu snadno vypočteme.

Konvergence Gaussovy – Seidelovy metody je zaručena pro pozitivně definitní matice a pro matice s převládající diagonálou (u matice s převládající diagonálou je každý diagonální prvek v absolutní hodnotě větší než součet absolutních hodnot ostatních prvků na stejném řádku). Pro matice s převládající diagonálou vždy konverguje i Jacobiho metoda. Ani v jednom případě však nejde o nutnou podmínku konvergence [55].

Řešíme-li soustavu  $\mathbf{A} \mathbf{x} = \mathbf{b}$ , začneme Gaussovu – Seidelovu metodu rozdělením matice  $\mathbf{A}$  na matici  $\mathbf{D}$  obsahující pouze její diagonální složky, matici  $\mathbf{L}$  představující část ležící pod diagonálou a  $\mathbf{U}$  část nad diagonálou. Soustavu pak přepíšeme takto:

$$(\mathbf{L}+\mathbf{D})\boldsymbol{x} = \boldsymbol{b} - \mathbf{U}\boldsymbol{x}$$
(2.3)

Předpokládejme, že diagonální prvky matice **A**, respektive **D**, jsou nenulové. Matice (L+D) je pak regulární a trojúhelníková a soustavu s ní umíme řešit rychleji než soustavu s původní rovnicí:

$$\mathbf{x}^{k+1} = (\mathbf{L} + \mathbf{D})^{-1} (\mathbf{b} - \mathbf{U} \mathbf{x}^{k})$$
 (2.4)

Tento vztah s doplněnými indexy iterací představuje iterační předpis Gaussovy – Seidelovy metody. Analogicky získáme vztah pro Jacobiho metodu, jen matici **A** rozdělíme na části **D** a (L+U):

$$\boldsymbol{x}^{k+1} = \boldsymbol{D}^{-1}(\boldsymbol{b} - (\boldsymbol{L} + \boldsymbol{U})\boldsymbol{x}^{k})$$
(2.5)

36
Obě metody lze interpretovat jako opakované přičítání vhodného násobku rezidua k aktuální aproximaci řešení; V případě Jacobiho metody lze psát  $\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{D}^{-1}(\mathbf{b} - \mathbf{A} \mathbf{x}^k)$ , u metody Gaussovy – Seidelovy je to  $\mathbf{x}^{k+1} = \mathbf{x}^k + (\mathbf{L}+\mathbf{D})^{-1}(\mathbf{b} - \mathbf{A} \mathbf{x}^k)$ . Vhodnou úpravou maticového koeficientu před reziduem lze rychlost konvergence ještě zvýšit. Toho využívá metoda známá jako *superrelaxační* (známá také jako SOR, ze slov *Successive Over-Relaxation*), která pracuje s předpisem  $\mathbf{x}^{k+1} = \mathbf{x}^k + \omega \ (\omega \mathbf{L}+\mathbf{D})^{-1} \ (\mathbf{b} - \mathbf{A} \mathbf{x}^k)$ . Reálný parametr  $\omega$  se volí mezi 0 a 2, a jeho optimální hodnota závisí na řešené úloze. Pro implementaci na FPGA má tato metoda nevýhodu ve vyšším počtu potřebných obvodů pro násobení obecných reálných čísel.

Výpočet jednoho kroku Jacobiho metody není algoritmicky (ani z hlediska FPGA) o mnoho jednodušší než u Gaussovy – Seidelovy metody, přestože není třeba řešit soustavu s trojúhelníkovou maticí, ale jen diagonální; To proto, že oproti metodě Gaussově – Seidelově je třeba víc operací při násobení vektoru maticí. Přitom se ukazuje, že Gaussova – Seidelova metoda konverguje rychleji [57]. Obě metody však mají společnou jednu vlastnost, která může být pro tomografii nevýhodná (ačkoli u multigridních metod je to důležitý předpoklad): Vysokofrekvenční složky řešení aproximují rychleji než nízkofrekvenční [68]. Pokud použijeme jako počáteční přiblížení konstantu, je nízkofrekvenční složkou řešení hrubý tvar profilu emisivity; Pokud vyjdeme z předchozího výsledku, může to být posunutí profilu, tedy věc důležitá pro případné využití ve zpětné vazbě. Je ovšem třeba mít na paměti, že analýza konvergence různých složek řešení se týká soustav vznikajících z diskretizace některých diferenciálních rovnic, a u tomografické úlohy se tyto metody mohou chovat jinak.

Gaussova – Seidelova a Jacobiho metoda se liší v možnostech paralelizace. U Jacobiho metody je paralelizace poměrně přímočará a patrná už z maticového zápisu, jako je (2.4). Násobení matice a vektoru o *n* řádcích lze provést například na *n* výpočetních jádrech, kde každé jádro počítá skalární součin vektoru s jedním řádkem matice. V sekvenční variantě výpočtu lze rovněž snadno využít řídkosti matice k optimalizaci tohoto kroku. Řešení soustavy s diagonální maticí spočívá ve vydělení každé složky pravé strany jím odpovídajícím diagonálním prvkem matice, což lze také provést odděleně. Naproti tomu v Gaussově – Seidelově metodě se vyskytuje soustava lineárních rovnic s trojúhelníkovou maticí, která se řeší zpětným dosazováním; Kvůli tomuto kroku nejsou složky řešení navzájem nezávislé, a v sekvenčním výpočtu se obvykle tyto složky počítají jedna po druhé. Nejspíš proto je občas paralelizace Gaussovy – Seidelovy metody považována za obtížnou [69].

Pokud je matice soustavy (a tudíž také trojúhelníková **L**+**D**) řídká, pak jednotlivé složky neznámé nezávisejí na všech předchozích složkách, ale jen na některých. Toho lze využít k rozdělení neznámých do skupin, v jejichž rámci jsou navzájem nezávislé, ale závisí na neznámých z některých dalších skupin. Neznámé ze stejné skupiny pak můžeme zpracovávat zároveň [69]. Pro rozdělení do skupin se využívají relativně složité (algoritmicky i výpočetně) algoritmy pro obarvení grafu, kde vrcholy grafu tvoří neznámé, navzájem závislé neznámé jsou spojeny hranou a každá barva výsledku odpovídá jedné nezávislé skupině. Pro řešení tomografické úlohy v reálném čase není složitost těchto algoritmů na překážku, protože rozmístění nenulových prvků řídké matice je pro všechny snímky stejné (mění se jen jejich hodnoty), a skupiny lze sestavit předem. Horším problémem je fakt, že efektivita této metody paralelizace silně závisí na řídkosti matice [69], a tomografická matice pro malé počty pixelů příliš řídká není (viz obrázek 2.2). Rovněž se zdá, že tento druh algoritmu není příliš vhodný

pro implementaci na FPGA (kvůli různorodosti využívaných postupů).

Nadějnější metoda paralelizace spočívá ve vhodné záměně pořadí operací během výpočtu. Nejjednodušší implementace funkce vykonávající jeden iterační krok Gaussovy Seidelovy metody obsahuje dva vnořené cykly a využívá jen jednu vektorovou proměnnou uchovávající aktuální aproximaci řešení; Ta je postupně přepisována novou aproximací. Kód 2 ukazuje mírně upravenou verzi tohoto algoritmu; Je zde použita pomocná vektorová proměnná a vnitřní cyklus je rozdělen na dvě části. Tato implementace se lépe převádí do paralelní podoby.

Druhá z funkcí definovaných v kódu 2.2 představuje jednu z několika možností paralelního vykonání toho samého algoritmu; Tato varianta byla posléze implementována do FPGA. Nejvýznamnější odlišností je záměna vnitřního a vnějšího cyklu; Díky tomu se v každém kroku nového vnitřního cyklu zapisuje do jiné složky vektoru řešení, zatímco pro čtení se přistupuje stále k té samé, x[j]. Vnitřní cyklus lze tedy provést v konstantním čase jako broadcast x[j] mezi výpočetní jednotky (pomocí sběrnice) a následný výpočet dvou aritmetických operací v každé jednotce. Vnější cyklus musí být v tomto případě rozdělen na dvě po sobě následující části, protože výpočty ve druhé části závisejí na výsledcích první části. Je zřejmé, že časová složitost takto implementovaného algoritmu je lineární v počtu rovnic, stejně jako počet využitých výpočetních jader. Nevýhodu má tento algoritmus v případě aplikace na řídké matice, protože neposkytuje možnost přeskočit výpočty s nulovými prvky; Nanejvýš je možné omezit počet výpočetních jader pro paralelní operace, má-li matice soustavy omezený počet nenulových prvků v každém sloupci. Rovněž je těžké dosáhnout další paralelizace, protože jednotlivé kroky druhé části vnějšího cyklu jsou sekvenčně závislé (lze uvažovat o blokové paralelizaci první části cyklu, ale pak bude tato část cyklu zabírat mnoho logických prvků, a časová složitost kroku zůstane lineární kvůli druhé části cyklu).



Obr. 2.2: Rozložení nenulových prvků (černě) v matici tomografické úlohy regularizované Fisherovou informací pro rozlišení 1210 pixelů (vlevo) a 104 pixelů (vpravo)

```
1 void gaussSeidel_sekvencni(double** A, double* b, double* x, double* x2, int n){
 2
       // A .. matice, b .. pravá strana, x .. předchozí aproximace řešení,
 3
       // x2 .. místo na novou aproximaci, n .. počet neznámých
 4
       // Inicializace pomocné proměnné hodnotami pravé strany
 5
 6
       for(int i=0; i<n; i++)x2[i] = b[i];</pre>
 7
       for(int i=0; i<n; i++){</pre>
 8
 9
            // Násobení horní trojúhelníkovou částí matice
10
            for(int j=i+1; j<n; j++){</pre>
11
                x2[i] -= A[i][j]*x[j];
12
            }
            // Zpětné dosazování
13
            for(int j=0; j<i; j++){</pre>
14
                x2[i] -= A[i][j]*x2[j];
15
16
            }
17
            x2[i] /= A[i][i];
18
       }
19 }
20
21 void gaussSeidel_paralelni(double** A, double* b, double* x, double* x2, int n){
22
       // Inicializace pomocné proměnné hodnotami pravé strany
23
       parfor(int i=0; i<n; i++)x2[i] = b[i];</pre>
24
       // První nezávislá část výpočtu
25
       for(int j=1; j<n; j++){</pre>
26
            parfor(int i=0; i<j; i++){</pre>
27
                x2[i] -= A[i][j]*x[j];
28
            }
29
       }
30
       // Druhá nezávislá část výpočtu
31
       for(int j=0; j<n; j++){</pre>
32
            x2[j] /= A[j][j];
33
            parfor(int i=j+1; i<n; i++){</pre>
34
                x2[i] -= A[i][j]*x2[j];
35
            }
       }
36
37 }
```

Kód 2.2: Dvě implementace Gaussovy Seidelovy metody; První z nich představuje přímočarý přepis vektorového vztahu do sekvenční podoby, ve druhé je pořadí operací změněno pro efektivnější paralelizaci. Jsou-li cykly **parfor** vykonány v konstantním čase, časová složitost paralelizovaného algoritmu je lineární.

#### 2.6 Paralelní ART

Metoda ART je efektivní ve své sekvenční podobě a nabízí i určité možnosti paralelizace, i když ty nejsou tak jednoduché jako u metod popsaných v předchozích odstavcích. Pro začátek lze uvažovat o paralelizaci jejího jednoho kroku. Ten ovšem zahrnuje výpočet skalárního součinu řádku matice s vektorem a kvadrátu normy řádku matice, tedy úlohy na paralelní redukci. Tyto úlohy lze vyřešit v logaritmickém čase na n výpočetních jádrech; U druhé z nich si lze pomoci jejím předpočítáním. Předpočítání lze provést efektivněji, totiž na n jádrech, z nichž každé (v lineárním čase) spočítá kvadrát normy jednoho řádku matice. To je třeba provést po každé změně matice, ale pokaždé to zabere jen lineární čas, zatímco počítat každou normu zvlášť by vyžadovalo čas  $O(n \log n)$ . Počítání skalárních součinů takto urychlit nelze, protože jeden

jejich operand se mění pořád. Výsledná časová složitost takto paralelizovaného algoritmu je tedy  $O(n \log n)$  na jeden iterační krok. Podobně se dá zřejmě paralelizovat i regularizovaná varianta ART, záleží ovšem na složitosti výpočtu gradientu hodnotícího funkcionálu.

Další (nebo alternativní, pro architektury, které se nehodí pro tak nízkoúrovňovou paralelizaci, jako je paralelní redukce sčítáním) paralelizace je ještě složitější, protože jednotlivé iterační kroky metody jsou obecně sekvenčně závislé. Každý krok vychází z mezivýsledku vyprodukovaného předchozím krokem a provádí s ním operaci promítnutí na nadrovinu definovanou jednou z rovnic soustavy. I tak je ovšem možné výpočet paralelizovat, protože některé z rovnic jsou ortogonální (a jim příslušné nadroviny jsou kolmé). To je umožněno tím, že v každé rovnici vystupuje jen omezená podmnožina neznámých, a lze najít dvě rovnice, které nemají žádnou neznámou společnou [70]. Provádíme-li po sobě dva iterační kroky s navzájem ortogonálními rovnicemi, oprava aplikovaná na mezivýsledek v prvním kroku nemá vliv na tvar opravy ve druhém kroku; Jde o posunutí kolmé na první nadrovinu a rovnoběžné s druhou. V běžném uspořádání zahrnujícím paralelní čárové projekce lze nalézt i větší skupiny ortogonálních rovnic, protože chordy z jedné paralelní projekce se neprotínají, a pokud jsou vzdáleny více než na šířku jednoho pixelu, pak není žádný pixel, kterým by jich procházelo více než jedna [70]. Díky tomu lze vypočítat všechny opravy příslušející stejné projekci zároveň, a pak je aplikovat na mezivýsledek pomocí paralelní redukce. Chordy vyskytující se v tomografii plazmatu bohužel nebývají paralelní ani ortogonální.

Lepší možnosti paralelizace poskytují některé modifikace metody ART, které lze s trochou fantazie všechny zařadit do třídy metod zvané *ordered subsets*. Odlišnost spočívá v tom, že počítáme více oprav mezivýsledku zároveň ("subset"), i když jejich rovnice nejsou ortogonální. Původním účelem ordered subsets bylo vhodným pořadím ("ordered") oprav zrychlit konvergenci algoritmu, a zároveň vhodným rozsahem skupin dosáhnout kompromisu mezi rychlou konvergencí a málo zašuměným výsledkem [71]. Výhoda z hlediska paralelních výpočtů je ovšem zřejmá.

Extrémním případem ordered subsets je metoda SIRT, zmíněná v předchozí kapitole; Spočítá nejprve opravy příslušející všem rovnicím, a mezivýsledek pak modifikuje všemi najednou. Výsledek tohoto postupu bývá málo zašuměný, ale dosáhnout jej trvá dlouho ve srovnání s metodou ART (kterou lze chápat jako opačný extrém mezi metodami typu ordered subsets). Přínos z možnosti paralelizace ovšem tuto nevýhodu do značné míry odstraňuje.

Metody typu ordered subsets mimo zmíněné dva extrémy bývají optimalizované pro rychlou konvergenci, a přitom se skládají z úseků výpočtů, které lze provádět paralelně. Příkladem takové metody je SART, v níž se nezávisle počítají opravy pro všechny rovnice příslušející jedné (paralelní) projekci. Oproti předchozím metodám je ovšem metoda SART doplněna o nerovnoměrné vážení opravy v různých pixelech ležících na chordě [16]. Ačkoli konvergence metody SART měřená počtem iteračních kroků je rychlejší než u metody SIRT, ukazuje se, že v implementaci pro grafické karty může být naopak rychlejší SIRT, protože umožňuje vyšší paralelizaci a méně časté změny ve schématu výpočtů (tedy střídání výpočtu a aplikace oprav) [72]. U FPGA ovšem může být situace odlišná.

### 2.7 Další implementační detaily

Rychlé výpočty na programovatelných hradlových polích se do značné míry podobají analogickým výpočtům prováděným na sekvenčním procesoru. I tak je ovšem třeba zavést mnohé další úpravy algoritmu kromě těch souvisejících s paralelizací. Především jde o to, že do FPGA se nevejde libovolně složitý design; Rozlišení paralelizované tomografie tomu musí být přizpůsobeno (v závislosti na implementaci). Chování tomografických rekonstrukcí přitom na rozlišení závisí, přinejmenším pro malé hodnoty rozlišení [6, 33]. U většiny modelů FPGA je omezena digitální reprezentace čísel (přinejmenším pokud chceme využít vestavěné násobičky a další vysokoúrovňové obvody). Reprezentace může být omezena na pevnou řádovou čárku, což představuje problém zejména z hlediska numerické stability algoritmů (v nepřesnosti vstupních dat žádný nový problém nenastává, protože ta vycházejí z AD – analogově digitálních – převodníků už v reprezentaci s pevnou čárkou; Rozsah vhodný pro výstup se dá alespoň předem odhadnout a polohu řádové čárky zvolit podle toho). Obvody pro plovoucí desetinnou čárku lze zkonstruovat i na zařízeních, která to nízkoúrovňově nepodporují, ale to vede k vyšší spotřebě logických prvků, případně i k nižší rychlosti násobení (pevné násobičky zvládnou jedno násobení za takt [73]).

Vstup dat je další součástí designu, kterou je třeba optimalizovat. Pokud se výpočet vykonává jednou za 50 µs, musí být se stejnou frekvencí načítána data ze všech vstupních kanálů, které odpovídají výstupním kanálům měkké rentgenové diagnostiky. Těch je na tokamaku COMPASS k dispozici 90 [10]. Prvním možným řešením situace je použít 90 paralelních AD převodníků se vzorkovací frekvencí nejméně 20 kS/s. Toto řešení skrývá několik komplikací. Pokud převodníky pracují s takto nízkou frekvencí, je dobré je synchronizovat s výpočetním cyklem v FPGA, aby mezi nimi a FPGA nevznikalo zpoždění; Lze ovšem použít převodníky s frekvencí vyšší. Dalším problémem může být omezený počet digitálních vstupů, kterými FPGA disponují (čip XC6SLX45 z řady Spartan 6 od firmy Xilinx jich má 358 [74], ovšem na desce použité k testování nejsou všechny přístupné), a také fakt, že každé implementované rozhraní pro AD převodník zabírá část logických elementů na čipu.

Opačnou možností je pro všechny vstupní kanály použít jeden (rychlý) AD převodník. Ten pak musí mít na vstupu *multiplexer*, tedy obvod, který k převodníku přivádí různé analogové vstupy na základě řídícího signálu. FPGA má pak pouze jeden vstup z AD převodníku (typický převodník potřebuje dva datové piny čipu) a jeden výstup pro multiplexer (pro přepínání 90 signálů je třeba 7 bitů, a tedy výstupních pinů). Převodník, který má zvládnout 90 převodů za 50 µs, musí mít frekvenci alespoň 1.8 MS/s (tedy podobnou, jakou mají převodníky používané u současných měkkých rentgenových detektorů na COMPASSu). Při použití minimální frekvence ovšem opět nastává problém: Některá data vstoupí do výpočtu jako aktuální, zatímco jiná budou 50 µs stará. Není přitom jisté, jak se s takto nekonzistentním vstupem zachovají tomografické metody; Pokud vyprodukují smysluplný výsledek, pak ještě není jisté, jakému času výsledek (respektive z něj odvozený údaj pro regulaci) odpovídá. Nejspíš je tedy vhodné použít převodník s vyšší frekvencí.

Logicky se ještě nabízí kompromis mezi jedním a devadesáti převodníky, kdy pracuje několik převodníků zároveň a každý z nich zpracovává několik signálů. Potřeba výstupních pinů pro multiplexery se tím nezvýší, protože pro všechny lze použít stejný řídící signál. Kvůli konzistenci dat je opět dobré použít převodníky s vyšší než minimální frekvencí, ovšem ta je teď o něco nižší než v případě jediného převodníku.

Konečně je možné omezit (už tak omezené) množství dat, které regularizovaná tomografická metoda potřebuje pro dostatečně kvalitní rekonstrukce. Jde o to, že data, která máme k dispozici, jsou určena konstrukcí vakuové komory a uspořádáním detektorů v jejích portech, a ne požadavky na rekonstrukci z omezeného množství dat. Lze tedy očekávat, že část vstupních dat je možné vynechat, přičemž kvalita výstupu se příliš nezhorší. Na podobné myšlence jsou založeny tomografické metody využívající *komprimovaného snímání (compressed sensing)*, jen se obvykle data vybírají z méně omezené množiny, a regularizace bývá formulována jinak [26].

V uvažované úloze není zřejmé, jak optimálně zvolit vstupní data, ačkoli u neregularizované tomografie lze očekávat, že dobré bude takové uspořádání detektorů, které pokryje projekční prostor co nejvíce rovnoměrně. Je tedy třeba použít metody určené pro *feature selection* (občas překládáno jako *výběr rysů*) nebo *feature extraction* (*extrakci rysů*) [75]. Metody z první skupiny vybírají z množiny dostupných datových vstupů ty nejvíce relevantní a s ostatními nepočítají, zatímco metody feature extraction obecně přepočítávají vstupní data na jiné veličiny, kterých je méně (například lineární kombinace původních vstupů). Je zřejmé, že z hlediska složitosti potřebné elektroniky zařazené před FPGA jsou výhodnější metody feature selection umožňuje lepší využití informace přítomné v datech.

Úkolem feature selection je z dostupné množiny datových kanálů vybrat podmnožinu o zadaném počtu prvků, která nejlépe splňuje určené kritérium (v našem případě nějak charakterizující kvalitu rekonstrukce). Množina o mohutnosti n má  $\binom{n}{k}$  podmnožin o

mohutnosti *k*, což ve většině případů vylučuje hledání optimální podmnožiny hrubou silou. Dá se ovšem využít toho, že kritérium je často možné vyčíslit i pro podmnožiny o mohutnosti jiné než *k*, a hledat podmnožinu postupným přidáváním prvků k výchozí prázdné množině nebo naopak odebíráním z úplné množiny. Pokud je zaručeno, že splnění kritéria se přidáním prvku k podmnožině může pouze zlepšit (nebo pouze zhoršit), lze prohledávání prostoru podmnožin urychlit metodou *branch and bound*, tedy vynecháním těch výpočtů, o nichž je známo, že nemohou přinést zlepšení oproti nejlepšímu dosud nalezenému výsledku. Bohužel ani taková optimalizace často nestačí, a je nutné se uchýlit k heuristickým metodám, i když ty nezaručují nalezení jediného optimálního řešení [76].

Jednou z nejjednodušších heuristických metod je hladový algoritmus, který začíná s prázdnou množinou a v každém kroku do ní přidá ten prvek, který přinese největší zlepšení (případně naopak). Vylepšením tohoto postupu jsou metody, které prvky střídavě přidávají a odebírají, dokud jich není správný počet. Tyto metody vyžadují více výpočtů kritéria, ale do jisté míry zabraňují hromadění prvků nesoucích redundantní informaci (pokud se přidáváním dalších prvků nějaký starší prvek stane redundantním, hladový algoritmus se ho neumí zbavit). Příkladem je metoda *plus-r-minus-l*, která pokaždé přidá konstantní počet prvků (*r*) a následně jiný počet prvků (*l*) zase odebere. I tyto metody ovšem patří mezi heuristiky [76].

Kromě výběru dat je třeba implementovat jejich jednoduché zpracování, které se provádí před samotnou tomografií. Ve většině testů v následující kapitole se počítá s daty, která jsou v čase shlazena klouzavým průměrem s oknem širokým 50 µs. Cílem této filtrace je omezit šum v datech a umožnit tak přesnější rekonstrukce. Pokud nechceme používat převodníky s významně vyšší než nezbytnou frekvencí a jejich výstupy filtrovat digitálně, můžeme si

pomoci analogovými low-pass filtry, které umístíme před převodníky (musí jich ovšem být 90, protože na ně nefungují triky s multiplexery). Přitom je třeba počítat se zpožděním, které může být filtrem způsobeno.

# 3 Testování algoritmů

V předchozích dvou kapitolách byla navržena řada možností, jak provádět rychlou tomografii plazmatu. Pro jednotlivé možnosti se navíc nabízejí různé způsoby časové optimalizace. Většina z výše popsaných metod byla implementována a výsledky jejich testování jsou v této kapitole. Cílem testování bylo nalézt algoritmus, který poskytuje věrohodné tomografické rekonstrukce v co nejrychlejším čase (dosažitelném na dostupném hardwaru), optimálně tak rychle, aby jej bylo možné použít ve zpětnovazebním řízení polohy plazmatu na tokamaku COMPASS.

K testování byla použita jak modelová data, tak data ze skutečných experimentů na tokamaku COMPASS. Modelová data byla získána aplikací operátoru projekce na fantomový profil emisivity. K těmto datům byl přidán umělý šum tak, aby byl výsledek náhodnou veličinou s Poissonovým rozdělením se střední hodnotou rovnou původnímu údaji. Odchylky tedy napodobují především kvantový šum, a jejich rozptyl je dán zvoleným přepočtem umělých signálů na počty fotonů. Testy s reálnými daty jsou nutné nejen proto, že skutečná emisivita plazmatu může mít jiné tvary, než s jakými počítá model, ale i kvůli ověření chování tomografických algoritmů na datech s rozmanitějšími chybami (zejména s chybami systematické povahy). Pro většinu pokusů s reálnými daty byly použity výboje v rozmezí #8696 až #8796, pro které byly k dispozici údaje ze všech tří instalovaných měkkých rentgenových kamer. Způsob získávání dat určuje, jakými kritérii může být měřena "věrohodnost" výsledků tomografie. U modelových dat byly výsledky nově implementovaných algoritmů porovnávány s modelem i s výsledky referenčních algoritmů aplikovaných na stejná data. U reálných dat není skutečný tvar profilu znám, ale zase jsou k dispozici magnetické diagnostiky, které poskytují informace o tvaru magnetického pole a poloze magnetické osy (i když tyto údaje mohou být zatíženy chybami).

Řadu z optimalizací určených k otestování lze navzájem kombinovat. Některé z nich přitom mají parametry, na jejichž hodnotách závisí úspěch optimalizace. Vzhledem k množství možných kombinací nebyly testovány všechny možnosti, nýbrž v některých případech byla správná kombinace odhadnuta obdobou hladového algoritmu, tedy po naladění první optimalizace byla aplikována další a předpokládalo se, že ta první nepřestane fungovat. Řada testů byla například prováděna s předem zvoleným rozlišením, ačkoli pro praktické použití se nakonec ukáže být vhodné jiné rozlišení.

### 3.1 Testy referenčního algoritmu

Jako reference byl použit tomografický algoritmus přímo řešící regularizovanou soustavu rovnic typu (1.11). Za hodnotící funkcionál byla zvolena Fisherova informace. Hodnota regularizačního parametru byla určována podle podmínky (1.12), tedy podle jedné z možných podob Morozovova principu. Hledání regularizačního parametru a Fisherových vah bylo prováděno ve stejném iteračním cyklu. Řešení přitom bylo považováno za nalezené, pokud se hodnota  $\chi^2$  z podmínky (1.12) odchylovala od požadované hodnoty nanejvýš o 5 % a zároveň v posledním iteračním kroku nedošlo v žádném prvku váhové matice ke změně větší než 2 %.

Za účelem porovnávání výsledků referenčního tomografického algoritmu s magnetickými

diagnostikami bylo implementováno několik algoritmů pro určování polohy (zejména vertikální) rekonstruovaného profilu. První z nich počítá těžiště části profilu, přičemž těžištěm se myslí vážený průměr poloh středů jednotlivých pixelů s hodnotami emisivity v roli vah. Část profilu pro výpočet těžiště je dána tím, které pixely překračují zvolený kvantil rozdělení emisivity; Tento kvantil představuje parametr algoritmu. Dalším vyzkoušeným algoritmem je hledání maxima profilu dodatečně vyhlazeného konvolucí s gaussovským jádrem (vyhlazování proto, aby se omezila citlivost na šum, který přežil regularizaci). Poslední metodou je fitování gaussovského profilu na profil emisivity, přičemž lze opět zvolit práh, pod nímž nebudou hodnoty do fitování započteny.

Metody pro určování vertikální polohy byly nejprve testovány na modelových datech, přičemž kritériem úspěchu byla podobnost mezi výsledkem metody aplikované na fantomový profil a výsledkem stejné metody aplikované na odpovídající rekonstrukci. Podobná statistika byla provedena s reálnými daty, kde za referenční údaj posloužila poloha určená magnetickými diagnostikami.

U modelových dat lze v odchylkách mezi referenční a testovanou hodnotou vertikální polohy plazmatu rozlišit systematickou a náhodnou složku. Vlivem systematické chyby se hodnoty neshodují, ale existuje mezi nimi závislost, s jejíž znalostí lze přepočítat výsledek získaný z tomografické rekonstrukce na původní hodnotu. Odstranit stochastickou odchylku takto nelze, protože ta je obecně při každém pokusu jiná, i když se poloha nemění. Pro potřeby řízení polohy v reálném čase je systematická odchylka méně nepříznivá než stochastická, protože ji lze opravit.

Odchylky produkované různými metodami byly analyzovány v rekonstrukcích fantomových profilů s využitím předpokladu, že systematickou chybu je možné napravit afinní transformací, tedy násobením polohy jednou konstantou a přičtením jiné. Jako testovací data byla použita sada 80 fantomů ve tvaru

$$f(R,z) = \exp\left(-\frac{(R-R_0)^2}{a^2} - \frac{(z-z_0)^2}{b^2}\right),$$
(3.1)

kde  $R_0$ ,  $z_0$  značí souřadnice středu profilu ( $R_0$  voleno mezi 0.48 m a 0.65 m a  $z_0$  mezi -0.15 m a 0.15 m) a parametry *a* a *b* určují šířku profilu (voleny nezávisle mezi 0.10 a 0.15 m); Byly vynechány takové profily, které se příliš blížily stěně komory. Analýza byla provedena pro čtyři druhy regularizačního funkcionálu (norma prvních derivací, norma druhých derivací a Fisherova informace s napůl pevnými okraji (viz podkapitola 1.5), Fisherova informace s pevnými okraji) a pro devět metod určení polohy plazmatu z rekonstruovaného profilu (těžiště emisivity s prahy odpovídajícími kvantilům 0, 0.25, 0.5 a 0.75, fitování se stejnými čtyřmi prahy a hledání maxima). Rekonstrukce byly prováděny pro každý regularizační funkcionál s rozlišením od pixelů o hraně 10 cm (komoru COMPASSu pokryje 38 takových pixelů) až po pixely o hraně 1 cm (3400 pixelů).

Prvním zjevným výsledkem testů je špatný výkon metody hledání maxima, který je zřejmě způsoben její omezenou rozlišovací schopností danou velikostí pixelu, a nejspíš i přetrvávající citlivostí na artefakty. Při nejvyšším rozlišení dosahuje metoda střední stochastické odchylky kolem 1.5 cm, jinak tato odchylka roste přibližně úměrně velikosti pixelu. Zůstává ovšem větší,

než by se dalo vysvětlit vlivem velikosti pixelů. Ostatní metody poskytují výsledky s nižší stochastickou chybou, které zřejmě závisejí spíše na volbě regularizace než na následném určování polohy.

Společnou vlastností většiny testovaných metod jsou vysoké odchylky (jednotky centimetrů) pro nízká rozlišení, odpovídající méně než přibližně dvěma stům pixelů. Tento problém je jasně patrný při použití lineárních regularizací i Fisherovy informace s pevnými okrajovými podmínkami. Naopak inverze regularizovaná Fisherovou informací s napůl pevnými konci má i při nejnižším rozlišení stochastickou odchylku menší než jeden centimetr; Systematická odchylka se ovšem pro malá rozlišení zvětšuje i u této metody. Při vyšších rozlišeních se stochastická odchylka už příliš nezmenšuje, a u většiny metod se ustálí na hodnotě mezi 0.5 a 0.8 cm (výjimku představuje regularizace normou druhých derivací, u níž se stochastická odchylka významně mění v celém zkoumaném rozsahu rozlišení). Systematická chyba pro rozlišení vyšší než několik stovek pixelů obecně nevykazuje sestupný trend, což je nejspíš dáno tím, že ji určuje nejen nedostatečný počet pixelů, ale i nedokonalost soustavy detektorů a tomografické metody. Vývoj systematické i stochastické odchylky při použití tří vybraných metod je znázorněn na obrázku 3.1. Bohužel není jasné, zda pozorování z tohoto odstavce představují obecné vlastnosti metod, nebo jestli to tak funguje jen u použitého typu fantomů. Rozdělení odchylky na systematickou a stochastickou je navíc určeno předpokládaným tvarem systematické složky, který nemusel být zvolen ideálně.

Odchylky polohy rekonstrukcí od fantomů jsou zřejmě rozdílné pro různé regularizační metody. Nabízí se otázka, jak se výsledky různých regularizací liší mezi sebou, případně zda je možné nahradit jednu regularizaci jinou (například použít lineární regularizaci místo nelineární nebo vybrat takovou metodu, která dává srovnatelně přesné výsledky při nižším rozlišení). Byly tedy vypočteny stochastické a systematické odchylky mezi výsledky čtyř uvažovaných metod regularizace. Analýza vycházela z rekonstrukcí s rozlišením 438 pixelů, při kterém už se u zkoumaných metod blíží stochastické odchylky nejlepší dosažitelné hodnotě. Vertikální polohy byly z těchto rekonstrukcí určeny jako těžiště hodnot překračujících medián profilu.

Tabulka 3.1 ukazuje naměřené stochastické i systematické odchylky. Z těchto údajů je možné odvodit několik závěrů. Za prvé, nejnižší vzájemnou střední stochastickou odchylku (2.4 mm) mají rekonstrukce regularizované první a druhou derivací; I jejich vzájemná systematická odchylka je malá. Pokud ovšem jde o stochastickou odchylku od fantomových dat, jsou na tom tyto metody hůře než obě testované nelineární regularizace. To naznačuje, že regularizace Fisherovou informací má vliv na rekonstrukci polohy plazmatu. Druhá nejnižší stochastická odchylka (3.8 mm) byla naměřena mezi oběma variantami regularizace Fisherovou informací. V porovnání s dvojicí lineárních regularizací se ovšem tyto metody liší: Fisherova informace s pevnými okraji má systematickou odchylku od první, respektive druhé derivace 7.5 mm a 9.3 mm; U Fisherovy informace s napůl volnými okraji je to jen 4.9 mm a 6.5 mm. To možná souvisí s faktem, že okrajové podmínky implementované v lineární regularizaci také odpovídají napůl volným koncům. Bohužel to ještě neznamená, že lze úspěšně nahradit Fisherovu informaci první derivací; Stochastická odchylka od fantomu je při použitém rozlišení patrně horší u první derivace.

Stochastické odchylky [mm]	Norma druhé derivace	Norma první derivace	Fisherova informace, pevné okraje	Fisherova informace, napůl pevné okraje
Fantom	8.1	7.1	5.9	4.7
Fisherova informace, napůl pevné okraje	6.5	4.9	3.8	
Fisherova informace, pevné okraje	9.3	7.5		-
Norma první derivace	2.4		-	

Systematické odchylky [mm]	Norma druhé derivace	Norma první derivace	Fisherova informace, pevné okraje	Fisherova informace, napůl pevné okraje		
Fantom	5.5	4.6	12.4	5.3		
Fisherova informace, napůl pevné okraje	10.3	7.8	8.4			
Fisherova informace, pevné okraje	13.2	10.8		-		
Norma první derivace	2.5		-			

Tab. 3.1: Stochastické (nahoře) a systematické odchylky vertikální polohy mezi rekonstrukcemi s různými regularizačními funkcionály a fantomovým profilem. Vertikální polohy byly měřeny na rekonstrukcích o 438 pixelech jako polohy těžiště z té části profilu, jejíž emisivita přesahuje medián.



Obr. 3.1: Systematická a stochastická složka odchylky vertikální polohy Δz v závislosti na rozlišení rekonstrukce. Poloha byla počítána jako těžiště nejjasnější poloviny profilu z rekonstrukcí umělých dat získaných metodami regularizovanými Fisherovou informací s pevnými okraji (vlevo), Fisherovou informací s napůl volnými okraji a normou druhé derivace s napůl pevnými okraji (vpravo).

Dále bylo provedeno srovnání poloh získaných z tomografických inverzí reálných dat s údaji z magnetických senzorů. Za tím účelem byly z dostupných experimentálních dat vybrány úseky z devíti výbojů (#8702, #8703, #8707, #8718, #8724, #8740, #8744, #8748, #8767, celkem 3940 časových snímků) s variabilní vertikální polohou. Data s původní vzorkovací frekvencí 2 MHz byla shlazena v čase konvolucí s obdélníkovým oknem o šířce 50 µs a se stejnou periodou převzorkována; Poloha určená z magnetických senzorů má také vzorkovací periodu 50 µs (jde o signál, podle kterého se řídí zpětná vazba vertikální polohy plazmatu). Pro rekonstrukce bylo opět zvoleno rozlišení 438 pixelů. Regularizační parametr byl určován na základě předpokladu desetiprocentního šumu v datech, což odpovídá relativní amplitudě té složky signálu, která má frekvenci vyšší než 1 kHz (byť samozřejmě není jisté, zda je to šum). Pro přepočet profilů emisivity na polohu byly opět vyzkoušeny algoritmy hledající těžiště části profilu nebo fitující Gaussovu křivku na část profilu; Hledání maxima bylo tentokrát vynecháno.

Výsledky analýzy odchylek prováděné na reálných datech se poněkud liší od výsledků získaných u fantomů. Na rozdíl od předchozího případu se ukazuje nezanedbatelná závislost odchylek na použité metodě hledání polohy; Ve většině případů (s výjimkou regularizace Fisherovou informací s pevnými okraji) se zdá, že metoda výpočtu těžiště dává nejlepší výsledky, pokud je aplikována jen na nejjasnějších 25 % pixelů, zatímco metoda založená na fitování nejlépe funguje s celým profilem. Vůbec nejnižší střední stochastickou odchylku od magnetického měření vertikální polohy (3.4, respektive 3.5 mm) dávají zmíněné algoritmy v kombinaci s regularizací normou první derivace. Střední systematická odchylka je ovšem v tomto případě velká (26.5 a 29.5 mm), zatímco nejmenší (7.2 mm) je při použití Fisherovy informace s pevnými okraji a výpočtu těžiště celého profilu. Všechny výsledky analýzy jsou zapsány v tabulce 3.2. Obrázek 3.2 znázorňuje dva příklady rekonstrukcí vertikální polohy v porovnání s magneticky měřenou polohou. V prvním případě (Fisherova informace s pevnými konci, těžiště jasnější poloviny profilu) se ukazuje špatný výkon afinní opravy systematické chyby, který naznačuje, že systematická chyba závisí na skrytých parametrech. Ve druhém případě (norma první derivace, těžiště nejjasnějších 25 % profilu) je rovněž patrná značná systematická chyba, ale její afinní oprava je úspěšnější. Přesto se však zdá, že u prvních dvou úseků dat (#8702, #8703) je systematická chyba jiná než u ostatních výbojů.

Věrohodnost získaných výsledků (výkon jednotlivých regularizačních funkcionálů a fungování navržených oprav systematických odchylek) byla testována s využitím dat odlišných od těch, ze kterých byly výsledky získány. Tentokrát byly použity úseky výbojů #8696, #8705, #8714, #8726, #8733, #8749, #8770 a #8784 (celkem 3000 snímků). Na těchto datech vyšla regularizace normou první derivace jako nejhorší, se stochastickými odchylkami překračujícími 10 mm při použití kterékoli metody pro hledání polohy. To naznačuje, že její dobrý výsledek na trénovacích datech byl náhodný (nastal jev označovaný jako *overtraining*). Regularizace Fisherovou informací s pevnými okraji si oproti tomu zachovala své poměrně dobré výsledky i na testovacích datech (stochastické odchylky kolem 6 mm, málo závislé na metodě měření středu). Použitelnost obou metod však ještě závisí na rychlosti jejich výpočtu; Může se ukázat, že málo přesné výsledky s vysokou frekvencí jsou užitečnější než opačná situace. Úplné výsledky měření na testovacích datech jsou opět v tabulce 3.2.

_					
	Fisherova informace, pevné okraje	Fisherova informace, napůl pevné okraje	Norma první derivace	Norma druhé derivace	
trénovací data					
těžiště 100 % profilu	6.1 / 7.2	6.7 / 16.2	7.6 / 18.5	10.8 / 14.9	
těžiště 75 % profilu	6.1 / 7.3	6.4 / 16.8	6.6 / 20.1	10.7 / 15.0	
těžiště 50 % profilu	6.0 / 7.6	6.0 / 18.1	5.2 / 22.9	10.4 / 18.6	
těžiště 25 % profilu	5.9 / 9.9	5.5 / 20.2	3.4 / 26.5	9.3 / 27.6	
fit 100 % profilu	6.5 / 15.4	5.2 / 24.7	3.5 / 29.5	7.6 / 27.6	
fit 75 % profilu	6.7 / 15.8	5.2 / 26.4	5.5 / 34.3	7.5 / 27.7	
fit 50 % profilu	6.1 / 15.0	5.9 / 24.1	5.3 / 32.7	10.1 / 32.5	
fit 25 % profilu	5.5 / 11.2	6.2 / 16.2	5.1 / 20.1	11.7 / 33.7	
testovací data					
těžiště 100 % profilu	6.2 / 4.4	8.1 / 13.2	32.9 / 14.6	10.5 / 25.2	
těžiště 75 % profilu	6.2 / 4.5	8.2 / 13.7	32.4 / 17.5	10.6 / 25.5	
těžiště 50 % profilu	6.2 / 4.8	8.0 / 15.0	35.0 / 17.9	11.0 / 28.9	
těžiště 25 % profilu	5.9 / 7.7	6.9 / 18.3	23.0 / 23.4	8.9 / 36.3	
fit 100 % profilu	5.7 / 14.8	6.4 / 23.4	35.2 / 30.4	8.3 / 44.0	
fit 75 % profilu	6.0 / 14.9	6.1 / 25.7	15.3 / 44.9	7.2 / 42.9	
fit 50 % profilu	5.7 / 13.5	6.4 / 24.0	10.8 / 34.5	7.4 / 41.4	
fit 25 % profilu	5.8 / 9.6	6.2 / 15.4	10.7 / 18.4	10.0 / 39.6	

Tab. 3.2: Stochastické / systematické odchylky vertikální polohy (v milimetrech) určené různými kombinacemi čtyř implementovaných tomografických metod a osmi metod pro určování polohy rekonstruovaného profilu; Bylo použito rozlišení 438 pixelů. Rozdělení na stochastické a systematické odchylky vzniklo fitováním afinní korekce na trénovací data, pro testovací data byly tyto korekce použity bez dalších úprav.



Obr. 3.2: Porovnání poloh plazmatu určených pomocí tomografie s údajem z magnetických diagnostik; Poloha v horním grafu byla vypočtena jako těžiště jasnější poloviny profilu získaného s Fisherovou regularizací, v dolním grafu jako těžiště nejjasnějších 25 % profilu regularizovaného normou první derivace. V obou grafech je zanesena i poloha získaná z tomografie se započtenou opravou systematické chyby.



Obr. 3.3: Ukázka rekonstrukce jednoho z fantomových profilů (vlevo) a reálného profilu z výboje #8717 v tokamaku COMPASS. Je zřejmé, že reálný profil je lépe lokalizovaný, než jak se předpokládalo v modelových datech. V obou případech byla použita regularizace Fisherovou informací s napůl pevnými okraji.

Výše popsané hodnoty odchylek, získané při rozlišení 438 pixelů, byly ověřeny i pro nižší rozlišení, konkrétně 306, 192 a 114 pixelů. V souladu s výsledky z fantomových dat nastává u některých metod významné zhoršení při 192 pixelech, zatímco při 114 pixelech už podává relativně dobré výsledky pouze regularizace Fisherovou informací s pevnými konci (nejnižší dosažená střední stochastická odchylka na testovacích datech 6.3 mm, u ostatních metod odchylky přesahují 10 mm). Statistika odchylek ovšem není ani po všech korekcích v dobré shodě s výsledky získanými na datech modelových. Jako možná vysvětlení se nabízí odlišný tvar a druh chyb v experimentálních datech (obrázek 3.3 ukazuje srovnání rekonstrukce fantomových a experimentálních dat), nespolehlivost referenčních dat z magnetických diagnostik, a nebo jen odlišný overtraining nastávající v obou případech. Jak u fantomových, tak u reálných dat lze však alespoň dojít k závěru, že regularizace Fisherovou informací poskytuje přesnější výsledky než obě vyzkoušené lineární regularizace. Na základě provedených testů se v dalších podkapitolách používá pro určování polohy algoritmus hledání těžiště nejvyšších 25 % profilu emisivity; Před fitováním byl upřednostněn pro svou rychlost a jednoduchost implementace.

#### 3.2 Optimalizace omezováním výpočtů

Není překvapivé, že výpočet tomografických rekonstrukcí s nižším rozlišením zabere kratší dobu. V prvním přiblížení se tedy vyplatí volit nejnižší rozlišení, při kterém je ještě možné získat dostatečně spolehlivé výsledky. Toto rozlišení lze zvolit na základě odchylek analyzovaných v předchozí podkapitole; Pokud použijeme regularizaci Fisherovou informací a požadujeme střední stochastickou odchylku od magnetických měření nižší než 10 mm, znamená to použít rozlišení kolem 114 pixelů. Pokud vyjdeme z analýzy fantomových dat, dojdeme k rozlišení 60 (Fisherova informace) až 200 (druhá derivace) pixelů. Tento výběr ovšem závisí na požadavcích uvažované aplikace.

Pro spolehlivější posouzení závislosti tvaru rekonstrukcí na rozlišení bylo navrženo kritérium nezávislé na metodách pro určování polohy; Jde o normu rozdílu rekonstrukce s malým rozlišením a referenční rekonstrukce s velkým rozlišením dělenou normou referenční rekonstrukce. Kompatibility rekonstrukcí s různým rozlišením bylo dosaženo bilineární interpolací zkoumaného profilu do rozlišení profilu referenčního (ačkoli přísně vzato nejde o rekonstrukce s bilineární interpolativní bází). Výsledky těchto testů naznačují, že rekonstrukce regularizovaná první derivací závisí na rozlišení méně než ta s Fisherovou informací. Bohužel není jasné, jestli to není dáno jen menší variabilitou lineární regularizace. Také se zdá, že pevná okrajová podmínka závislost na rozlišení snižuje, alespoň u Fisherovy informace, kde byla testována. Toto pozorování je v souladu s předchozími testy, které ukazují, že systematická odchylka polohy se u Fisherovy informace s pevnými okraji s rozlišením mění mezi 300 a 3000 pixely v desítkách procent, zatímco u Fisherovy informace s napůl pevnými konci se změní asi trojnásobně; Stochastická odchylka je ovšem poměrně stabilní u obou metod. Zajímavým pozorováním je také fakt, že ani u jedné z metod se nevyskytuje práh rozlišení, nad kterým by rekonstrukce na rozlišení závisela výrazně méně než pod ním. Výsledky jsou znázorněny na obrázku 3.4.



Obr. 3.4: Závislost relativní normy rozdílu rekonstrukce a referenčního řešení Δg na velikosti pixelů Δx pro různé regularizační funkcionály. Pro výpočet normy rozdílu byla zkoumaná rekonstrukce interpolována do stejného rozlišení, jaké měla rekonstrukce referenční. Ta byla vždy počítána se stejnou regularizací a s nejmenšími pixely ze zkoumaného rozsahu (tj. pixely o hraně 9 mm).

Pro posouzení užitečnosti snižování rozlišení byla změřena rychlost výpočtu tomografických inverzí pro různá rozlišení. Vzhledem k tomu, že dosud není zřejmé, jaký počet iterací je potřeba pro zpětnou vazbu (výše zmíněné kritérium konvergence je míněno spíše jako bezpečné pro referenční algoritmus než optimální pro řízení), údaje zaznamenané v obrázku 3.5 odpovídají trvání jednoho iteračního kroku. Algoritmus byl implementován v jazyce MATLAB. Kritická část výpočtu úlohy regularizované Fisherovou informací je řešena Choleského faktorizací [77] s použitím optimalizací pro řídké matice, zatímco u lineární regularizace je použita metoda GSVD. Výpočet běžel na procesoru Intel<sup>®</sup> Core<sup>™</sup> 2 Duo E6550 s frekvencí 2.33 GHz.

Měření výpočetních časů u Fisherovy informace ukazuje na exotickou časovou složitost  $O(n^{1.5})$ ; Tato složitost je ovšem poněkud nižší než  $O(n^3)$ , jakou by měla konvenční Choleského faktorizace. Není bohužel jasné, proč implementace dosahuje právě takovéto složitosti. Z obrázku 3.5 je patrné, že ani výpočet jedné iterace při nejnižším z testovaných rozlišení (38 pixelů) se nevejde do 50µs deadlinu. Ukazuje se přitom, že zatímco při vysokých rozlišeních zabírá řešení lineární soustavy 90 % výpočetního času, při nízkém rozlišení je to kolem 50 %, a zbývající polovinu zabere sestavování matice úlohy a výpočet rezidua; Zdá se tedy, že pro případnou aplikaci v reálném čase nestačí optimalizovat jen tuto kritickou část výpočtu.

Lineární regularizace je výhodnější nejen v tom, že pro řešení soustavy lze použít GSVD, ale také proto, že není třeba sestavovat pokaždé novou matici. Její výpočet by měl mít složitost  $O(n^2)$ ; Oproti předchozí metodě není třeba invertovat řídkou matici, zato je třeba dvakrát násobit vektor plnou maticí (maticí inverzní k **Z** a **Z**<sup>T</sup> ze vztahu 1.15). Naměřená závislost výpočetních časů tentokrát nemá jednoznačný polynomiální charakter, ovšem kvantitativně lze pozorovat, že ve zkoumaném rozsahu rozlišení je výpočet dvakrát (pro desítky pixelů) až



Obr. 3.5: Změřená trvání výpočtu jednoho iteračního cyklu v závislosti na rozlišení rekonstrukce. Řešení soustavy lineárních rovnic bylo u nelineárně regularizované úlohy provedeno Choleského transformací optimalizovanou pro řídké matice, u lineárně regularizované úlohy metodou GSVD.

čtyřikrát (pro tisíce pixelů) rychlejší než řešení soustavy bez GSVD. Zejména pro nízká rozlišení tedy lineárně regularizovaná tomografie není o mnoho rychlejší než ta s Fisherovou informací.

Z předchozí analýzy vyplývá, že se vyplatí použít co nejnižší rozlišení, které ještě dává smysluplné výsledky; Ani tak patrně není zaručeno, že výpočet splní rozumný deadline. Je zřejmé, že u všech druhů regularizace je rovněž nutné co nejvíce omezit počet iterací, aby bylo možné uvažovat o využití metody v reálném čase; Při nižším počtu iterací však patrně budou výsledky méně přesné. Byla tedy proměřena konvergence údaje o vertikální poloze v závislosti na počtu iterací algoritmu, jenž zároveň hledá lineární přiblížení Fisherovy informace a optimální hodnotu regularizačního parametru; Pro srovnání byla provedena stejná statistika i s úlohou regularizovanou první derivací. Byla použita stejná experimentální data jako pro analýzu náhodných a systematických odchylek. Jako výchozí stav pro iteraci byla zvolena hodnota regularizačního parametru 1 a jako počáteční přiblížení rekonstrukce konstantní funkce o hodnotě 1 (toto přiblížení je ovšem využito jen v nelineární regularizaci).

Výsledky ukazují, že už po první iteraci obou metod se zvoleným počátečním stavem se vertikální poloha určená z rekonstrukcí od limitní polohy liší v průměru o méně než sedm milimetrů (ačkoli rekonstrukce proběhla s regularizačním parametrem rovným jedné, přičemž optimální hodnota u použitých dat se obvykle pohybuje mezi 1 a 5). Během následujících šesti



Obr. 3.6: Závislost střední stochastické odchylky vertikální polohy na počtu iterací prováděných při rekonstrukci každého snímku. Jako počáteční přiblížení byla použita konstanta (horní dva grafy), respektive výsledek rekonstrukce předchozího časového kroku (opět s daným počtem iterací). Lze si povšimnout, že odchylka měřená po jedné iteraci s regularizací Fisherovou informací (vpravo nahoře) se neshoduje s odchylkou jedné iterace regularizované normou první derivace, ačkoli v tomto prvním kroku jde o stejné výpočty; Jde o to, že tyto mezivýsledky jsou porovnávány s rozdílnými referenčními výsledky, kterými jsou výstupy obou metod po čtyřiceti iteracích.

(u první derivace) až čtrnácti (u Fisherovy informace) iterací se pak odchylka zmenší pod 1 mm. Následný vývoj odchylky u Fisherovy informace se zdá být pomalejší než u první derivace, ale to už jde jen o desetiny milimetru. Dalším zajímavým rozdílem je fakt, že pokles odchylky u první derivace je monotonní, kdežto u Fisherovy informace bylo v prvních několika krocích zaznamenáno zhoršení. Naměřené odchylky v závislosti na počtu iterací jsou zakresleny v obrázku 3.6.

V práci [35] bylo navrženo snížit počet iterací tomografického algoritmu tím, že jako počáteční přiblížení použijeme stav získaný v předchozím časovém kroku. To může být poněkud

efektivnější než začít s konstantním profilem emisivity a s regularizačním parametrem rovným 1, pokud lze předpokládat, že profily emisivity naměřené v krátkém čase po sobě jsou podobné. S využitím tohoto triku byl proveden stejný test jako v minulém odstavci, tedy byla měřena závislost polohy rekonstruovaného profilu na počtu provedených iterací při regularizaci Fisherovou informací a normou první derivace. Tentokrát dosahuje lineární regularizace již při jedné iteraci průměrných odchylkek nižších než 1 mm, u nelineární jsou to 3 mm. Tyto odchylky se s rostoucím počtem iterací snižují, u nelineární regularizace opět o něco pomaleji; Změřené závislosti odchylek na iteracích jsou znázorněny na obrázku 3.6. Práce [35] navrhuje s použitím této metody snížit počet iterací u nelineární regularizace na jednu a řádově se tak přiblížit požadavkům na zpětnovazební řízení polohy na tokamaku COMPASS (tento postup je v dalším textu označován jako *klouzavá iterace*). Provedené měření naznačuje, že by tím nemělo dojít k průměrnému zhoršení přesnosti určení polohy o více než tři milimetry.

Ačkoli použitím vhodného počátečního přiblížení bylo dosaženo značného zlepšení v průměrné stochastické odchylce po daném počtu iterací, ukazuje se, že tato odchylka není mezi snímky rozložena zcela stochasticky, nýbrž že je vyšší u snímků následujících po rychlé změně polohy plazmatu (jako je střih v testovacích datech). Toto chování bylo následně analyzováno s využitím fantomových dat, která obsahovala dvě fáze stacionárního profilu emisivity oddělené skokem polohy o zvolené velikosti. U algoritmu s Fisherovou informací, využívajícího tvaru předchozí rekonstrukce, se ukazuje, že vertikální poloha rekonstrukce se k nové poloze fantomového profilu exponenciálně přibližuje, a to s časovou konstantou odpovídající 2.1 ± 0.4 snímkům (viz obrázek 3.7). Exponenciální útlum se zdá být nezávislý na tom, jak velký byl počáteční skok fantomové polohy, což naznačuje, že chybně rekonstruovaný profil se podobá profilu rekonstruovanému správně, avšak jinak umístěnému. To zřejmě souvisí s tím, že metoda upřednostňuje nalezení vrcholu emisivity v místě, kde byla emisivita vysoká i v minulém snímku.



Obr. 3.7: Časový vývoj údaje o vertikální poloze určené tomografickou rekonstrukcí fantomových dat obsahujících pravidelné skoky polohy; V obou případech byla odstraněna systematická odchylka. U signálu získaného s pomocí nelineární regularizace je po každém skoku patrná postupná konvergence k referenční hodnotě (až na šum). Podle očekávání nebylo exponenciální chování chyb u lineární regularizace pozorováno, protože u ní jsou po sobě jdoucí snímky provázány jen hodnotou regularizačního parametru. Rovněž bylo pozorováno zkrácení časové konstanty útlumu chyby polohy, pokud bylo pro každý snímek provedeno více než jedna iterace algoritmu s Fisherovou informací: Při dvou iteracích byla změřena konstanta  $1.3 \pm 0.4$  snímků, při třech  $0.7 \pm 0.2$  snímků. Vzhledem k tomu, že výpočet dvou či tří iterací trvá dvakrát, respektive třikrát déle, konstanta měřená v sekundách zůstává přibližně stejná. Ačkoli to nebylo ověřeno, lze očekávat, že podobný problém nastane i tehdy, pokusíme-li se použít jednou rozloženou matici pro více časových snímků, jak bylo navrženo v podkapitole 1.6.

Vzhledem k systematickému charakteru exponenciálně klesající chyby můžeme uvažovat o její opravě. Chybu lze (ve spojitém i diskrétním případě) modelovat jako filtraci signálu udávajícího správnou polohu pomocí filtru s nekonečnou impulzní odezvou:

$$z_{m \, \tilde{e} \, \tilde{r} e n \, \hat{q}}(t) = z(t) * (\exp(-t/\tau) \theta(t)). \tag{3.2}$$

Zde hvězdička značí konvoluci a  $\tau$  časovou konstantu útlumu chyby;  $\theta$  je jednotkový skok, tj. funkce nabývající hodnoty 0 pro záporné argumenty a 1 pro kladné. Vliv této filtrace analyticky snadno odstraníme další konvolucí:

$$z_{m\check{e}\check{r}en\acute{a}}(t)*(\delta'(t)+\tau^{-1}\delta(t)) \propto z(t). \tag{3.3}$$

Je ovšem zřejmé, že tento inverzní filtr představuje horní propust (díky členu s derivací Diracovy funkce  $\delta$ ) a jeho použitím dojde ke zvýraznění šumu v údaji o poloze. Popsaná korekce byla ověřena na stejných modelových datech, na jakých byly měřeny časové konstanty útlumu chyb; Přiblížení diskrétního inverzního filtru bylo implementováno jako konvoluce s jádrem ( $\tau/2$ ,  $1-\tau/2$ ). Tato operace vskutku redukovala pozorované exponenciální závislosti natolik, že nebyly rozpoznatelné oproti náhodnému šumu. Směrodatná odchylka šumu v úsecích stabilní polohy se přitom zvýšila asi na dvojnásobek. Obrázek 3.8 představuje ukázku původního polohového signálu v porovnání s opraveným. Zůstává otázkou, zda popsaná oprava může být užitečná pro skutečné zpětnovazební řízení polohy, případně jaký by byl její přínos v kombinaci s algoritmem, jehož jedno vlákno rozkládá matici a druhé aplikuje poslední dostupný rozklad matice na data.

#### 3.3 Použití iteračních algoritmů

Ve všech předchozích testech se počítalo s přímým řešením regularizované soustavy rovnic Choleského faktorizací nebo jiným druhem rozkladu matice. Může se ovšem ukázat, že některá z iteračních metod může být k tomu účelu vhodnější, ať už z hlediska implementace na dostupném hardwaru nebo díky své větší variabilitě. Za účelem ověření vlastností iteračních metod byly implementovány dvě z nich, Gaussova – Seidelova metoda a metoda konjugovaných gradientů. U těchto metod byl zkoumán výkon na datech napodobujících reálný čas (se vzorkovací frekvencí 50 µs), a to v závislosti na počtu iterací použitých k výpočtu jednoho snímku. Dá se předpokládat, že při krátkém deadlinu se stihne méně iterací s danou sadou dat, ale tato data jsou více podobná minulému vstupu; Frekvence iterací závisí jen na implementaci a na hardwaru a s deadlinem se nemění. Cílem testů tedy bylo především zjistit, kolik iterací je třeba provést za padesátimikrosekundový interval, případně která metoda jich potřebuje nejméně. Testy byly prováděny na úloze regularizované Fisherovou



Obr. 3.8: Vývoj polohy získané nelineárně regularizovanou rekonstrukcí fantomových dat se skoky polohy v porovnání se stejným signálem po aplikaci filtru omezujícího zpoždění.

informací s první ze dvou sad experimentálních dat zmíněných výše. Jako kritérium úspěchu byl tentokrát použit normalizovaný rozdíl profilu od výstupu referenčního algoritmu řešícího stejně regularizovanou úlohu se stejným rozlišením (bez klouzavé iterace), aby se omezil vliv dalších nepřesností v tomografické úloze a v určování polohy.

Algoritmus je možné implementovat ve značném počtu podob, s různými volbami iterační metody, s různě častým přepočítáváním váhové matice a různě zvolenými počty iterací. Pro zjednodušení testů se počítalo s konstantním rozlišením 438 pixelů. Takto byla nejprve vyzkoušena Gaussova – Seidelova metoda a metoda konjugovaných gradientů za předpokladu, že váhová matice se přepočítává zároveň s načtením nových dat (to odpovídá výše popsané klouzavé iteraci, v níž je přesné řešení soustavy lineárních rovnic nahrazeno iteračním výpočtem s omezeným počtem kroků). Prvním patrným výsledkem je fakt, že konstantní přiblížení řešení, jaké je běžně používáno k inicializaci iterační minimalizace Fisherovy informace, se nehodí pro inicializaci Gaussovy – Seidelovy metody ani metody konjugovaných gradientů. U obou metod zůstává konstantní složka dlouho patrná, ačkoli řešení konverguje ke tvaru s jasně odděleným vrcholem a oblastí téměř nulové emisivity. Konstantní počáteční přiblížení použit výsledek z předchozího času.

Dalším pozorováním je špatný (ve srovnání s Gaussovou – Seidelovou metodou) výkon metody konjugovaných gradientů s jakýmkoli počátečním přiblížením. To může souviset s citlivostí metody konjugovaných gradientů na špatnou podmíněnost soustav [78]; I po regularizaci totiž matice soustavy mívá některá vlastní čísla o čtyři řády menší než jiná. Aby se výsledky testu nezakládaly jen na použití jedné iterační metody, byla implementována a do testu zařazena metoda *předpodmíněných konjugovaných gradientů*, která místo soustavy v původním tvaru **A**  $\mathbf{x} = \mathbf{b}$  řeší ekvivalentní soustavu  $\mathbf{E}^{-1} \mathbf{A} \mathbf{x} = \mathbf{E}^{-1} \mathbf{b}$ , kde ( $\mathbf{E} \mathbf{E}^{T}$ ) je vhodně zvolená regulární matice [56]; V tomto případě byla použita diagonální matice s diagonálními prvky shodnými s maticí **A** (*Jacobiho preconditioner*). Výpočet iteračního kroku předpodmíněné metody je složitější a ještě méně vhodný pro implementaci na FPGA (zahrnuje více dělení), ale metoda má poněkud vyšší rychlost konvergence než ta bez předpodmínění.

Tabulka 3.3 shrnuje výsledky testů konvergence tří uvažovaných iteračních metod pro různé volby počátečního přiblížení; Konvergence byla sledována, dokud se průměrná hodnota normy rozdílu obrazů nesnížila pod 5 %. Je zjevné že nulové počáteční přiblížení je lepší než nenulová konstanta, a inicializace předchozím výsledkem je ještě lepší. Pro nulové počáteční přiblížení se nejlépe osvědčuje předpodmíněná metoda konjugovaných gradientů, ovšem i Gaussova – Seidelova metoda dosáhne průměrné odchylky pod 5 % do deseti iterací. Výpočet deseti iterací obou těchto metod je přitom poněkud rychlejší než přímé řešení soustavy o 438 rovnicích. Při použití posledního výsledku je výkon obou metod srovnatelný, a zdá se, že již po několika iteracích dosáhnou optimálního výsledku (optimálním výsledkem není nulová odchylka, protože referenční algoritmus počítá s iterativním zpřesňováním Fisherovy informace, kdežto doposud testované algoritmy jen řeší soustavu se stále stejným přiblížením hodnotícího funkcionálu). Obrázek 3.9 ukazuje příklad profilu rekonstruovaného referenčním algoritmem a klouzavou iterací s Gaussovu – Seidelovou metodou omezenou na pět kroků; Je zjevné, že druhá rekonstrukce ani přes značné zjednodušení výpočtu neobsahuje významné artefakty.

Kromě konvergence samotných iteračních metod pro řešení soustavy lineárních rovnic byly otestovány dvě varianty iteračního postupu s častějším přepočítáváním váhové matice; V prvním případě byla iterace Gaussovy – Seidelovy metody mezi každými dvěma vstupy dat dvakrát přerušena přepočtem matice, ve druhém případě třikrát. Konvergence metody tím narušena nebyla, naopak bylo dosaženo zlepšení měřené odchylky na 2.8 %, respektive 2.3 %, i při použití nejradikálnějšího přístupu, při němž se mezi přepočty matice stihne jen jedna iterace Gaussovy – Seidelovy metody. Optimální vyvážení frekvence přepočtů váhové matice a Gaussových – Seidelových iterací ovšem závisí na konkrétních experimentálních podmínkách, zejména na rychlosti implementace algoritmu a na požadavcích na přesnost.

Lze vytušit, že průběh chyby ve výše popsaných případech bude opět nerovnoměrný, protože jde o klouzavou iteraci, a při použití předchozího výsledku k inicializaci iteračního řešení soustavy jsou po sobě jdoucí snímky závislé ještě více. Toto podezření potvrzuje obrázek 3.10. Podle porovnání vývoje odchylky u Gaussovy – Seidelovy metody inicializované nulami a předchozím výsledkem se přitom zdá, že tato inicializace nemá na útlum odchylky velký vliv, a jde zejména o konvergenci regularizačního parametru  $\lambda$ , který je počítán metodou regula falsi. Není ovšem zřejmé, zda konvergenci  $\lambda$  neovlivňuje použití omezeně přesné iterační metody. Kvůli nepřesnosti výsledku totiž může být nepřesný i výpočet Morozovova kritéria, a to systematicky (například se předběžné výsledky mohou zdát nedostatečně vyhlazené, ačkoli po delší iteraci s daným  $\lambda$  by skonvergovaly ke správnému tvaru). Rozdíly v konvergenci vysokofrekvenčních a nízkofrekvenčních složek řešení byly zkoumány pro metodu předpodmíněných konjugovaných gradientů a Gaussovu – Seidelovu metodu, obě inicializované nulovým vektorem. U metody Gaussovy – Seidelovy se vskutku zdá, že dochází k rychlé eliminaci vysokofrekvenčních odchylek, zatímco hladká složka odchylky se zmenšuje pomaleji. Regularizace dosažená omezením iterací této metody by tedy byla spíše



Obr. 3.9: Porovnání rekonstrukcí snímku z výboje #8707 pomocí referenčního algoritmu (vlevo) a pomocí klouzavé iterace využívající Gaussovy – Seidelovy metody omezené na pět kroků; Gaussova – Seidelova metoda byla inicializována rekonstrukcí předchozího snímku, vypočteného stejnou metodou. Obrázek vpravo představuje rozdíl mezi oběma rekonstrukcemi (v barevné škále odpovídající 4 % škály absoluních snímků).



Obr. 3.10: Časový vývoj relativní normy rozdílu snímků rekonstruovaných s využitím iteračních metod od referenční rekonstrukce (s Choleského faktorizací); Znázorněny jsou výsledky pro Gaussovu – Seidelovu metodu omezenou na pět kroků, inicializovanou nulami (vlevo nahoře) a předchozím výsledkem (vpravo nahoře), předpodmíněné konjugované gradienty omezené na pět kroků a inicializované předchozím výsledkem (vlevo dole) a tři opakování předchozím výsledkem inicializované Gaussovy – Seidelovy metody omezené na dva kroky, přičemž před každým opakováním je přepočtena matice regularizované úlohy. Svislé čáry označují střihy v datech.

kontraproduktivní, protože lokální artefakty by zůstaly zachovány, zatímco tvar profilu by mohl být poškozen. V praxi je ovšem konvergence všech složek poměrně rychlá, použijeme-li rozumné počáteční přiblížení (viz tab. 3.3). Při použití předpodmíněných konjugovaných gradientů žádný podobný trend pozorován nebyl, spíše se zdá, že vysokofrekvenční i nízkofrekvenční odchylky se snižují podobnou rychlostí. Popsané chování ilustruje obrázek 3.11.

Na závěr bylo ověřeno, zda malé normě rozdílu mezi dvěma rekonstrukcemi odpovídá i malá vzájemná odchylka vertikálních poloh. Mezi polohami těžišť nejjasnějších 25 % profilu byla naměřena střední stochastická odchylka 4.8 mm a systematická 1.1 mm. Stochastická odchylka od polohy měřené magnetickými senzory činí 4.8 mm, což je dokonce lepší než u referenčního algoritmu (5.9 mm). To možná souvisí s tím, že v poloze získané z výstupu klouzavé iterace je méně šumu; Jak už bylo ukázáno, závislost mezi snímky působí na měřenou polohu jako dolní propust.

metoda →	GS	CG	PCG	GS	CG	PCG	GS	CG	PCG	2×GS	3×GS
inicializace $\rightarrow$	kon	kon	kon	0	0	0	last	last	last	last	last
iterace ↓	norma rozdílu [%]										
1						70.9	4.1	306.5	4.2	2.8	2.3
2						21.9	3.8	86.9	3.8	2.6	2.1
3				318.0		14.2	3.9	58.0	3.7	2.6	2.1
4				10.5		10.7	3.7	45.1	3.7		
5				8.4		7.7	3.6	34.9	3.6		
7				5.7		3.8		19.4			
10				3.5				13.6			
15			169.4					9.6			
25	253.6		46.6		63.4			7.3			
35	190.3		3.8		32.9			6.1			
50	103.9	343.3	3.6		12.0			5.2			
100	3.6	112.7			3.7			4.1			
200		38.2									
400		6.0									

Tab. 3.3: Konvergence tomografie založené na iteračních metodách měřená relativní normou rozdílu oproti výstupu referenčního algoritmu. Byly použity metody Gaussova – Seidelova (GS), konjugované gradienty (CG) a předpodmíněné konjugované gradienty (PCG). Pro metodu GS byla vyzkoušena i dvoj- a trojnásobná varianta, u nichž "n iterací" znamená dvě, respektive tři opakování procesu zahrnujícího výpočet váhové matice a n kroků GS metody. Konvergence byla měřena pro tři druhy počátečního přiblížení: konstantní přiblížení o předem odhadnuté hodnotě (kon), nulové počáteční přiblížení (0) a inicializaci výsledkem rekonstrukce předchozího snímku (last).



Obr. 3.11: Tvar odchylky od referenčního řešení po jedné až deseti iteracích Gaussovy – Seidelovy metody (horní řada) a předpodmíněných konjugovaných gradientů. Obě metody byly inicializovány nulovým přiblížením. Barevná škála jednotlivých snímků se liší.

#### 3.4 Implementace na FPGA

Tomografický algoritmus založený na Tichonovově regularizaci byl v několika variantách implementován v jazyce VHDL. Design byl navržen pro čip XC6SLX45 z řady Spartan 6 od firmy Xilinx a testován na desce Atlys vybavené tímto čipem. Čip má 6822 logických bloků, 58 osmnáctibitových násobiček pro pevnou řádovou čárku a 116 paměťových bloků o kapacitě 18 kb (1024 osmnáctibitových slov) [74]; Pracuje s hodinovým signálem o frekvenci 100 MHz. Většina testů byla prováděna v simulaci, zařízení bylo použito jen k ověření, zda chování obvodu simulaci odpovídá (rozdíly mohou vzniknout například zanedbáním požadavků na timing).

Navržený obvod se skládá ze čtyř hlavních komponent; Jedna z nich sestavuje po řádcích matici regularizované úlohy, druhá řeší vzniklou soustavu lineárních rovnic, třetí ze získaného řešení počítá hodnotu  $\chi^2$  podle vztahu (1.12). Poslední komponenta obsahuje stavový automat synchronizující činnost ostatních modulů, organizuje tok dat a provádí několik zbývajících jednoduchých operací (výpočet hodnoty  $\lambda$  na základě  $\chi^2$ , přístup k blokovým pamětem u metod, které jich mohou využít, výpočet inverzních hodnot diagonálních prvků matice pro optimalizaci běhu Gaussovy – Seidelovy metody).

Pro aplikaci ve zpětné vazbě polohy plazmatu v tokamaku by bylo nutné doplnit zmíněné komponenty o ovladače vstupních AD převodníků, přepočet tomografické rekonstrukce na hodnotu polohy (případně přímo signál pro řízení aktuátorů) a modul zajišťující výstup dat (například přes DA převodník nebo sériovou linku). Implementace těchto částí designu ovšem závisí na podobě zbytku zpětnovazební smyčky, který v současnosti na tokamaku COMPASS není k dispozici. Místo toho byla tedy implementována dvě rozhraní pro testování designu. Jedno z nich je určeno pro simulaci na konvenčním počítači; Načítá data z textového souboru v připraveném formátu, v němž každý řádek odpovídá výstupu všech dostupných detektorů v daném okamžiku, a v analogické podobě vypisuje výsledky. Tyto operace jsou rovněž implementovány v jazyce VHDL, ovšem nejsou *syntetizovatelné*, tedy dají se použít jen v simulacích [50]. Druhé rozhraní je vhodné pro testy v reálném hardwaru. Obsahuje sadu předem připravených vstupních dat a hash očekávaného výsledku (výsledek má podobu vektoru osmnáctibitových slov, výpočet hashe spočívá v redukci všech těchto slov bitovým xor).

Příprava matice regularizované úlohy spoléhá na konstanty zapsané přímo ve zdrojovém kódu příslušné komponenty, a tím pádem uložené v distribuované paměti čipu. Podobně je řešen výpočet Morozovova kritéria, při němž je třeba násobit vektor projekční maticí. Vzhledem k charakteru kódu realizujícího tyto výpočty byly implementovány skripty v jazyce Python, které načítají projekční a regularizační matice vygenerované MATLABem a generují VHDL. Nový kód je třeba vygenerovat po každé změně geometrie detektorů, rozložení a počtu pixelů. Ačkoli implementace zmíněných dvou modulů nevyužívá blokových pamětí, lze nahlédnout, že využívá řídkosti projekční a regularizační matice; Jde o tak jednoduché operace, že jejich optimalizace pro řídké matice nevede ke kritickému zesložitění designu. Generování matice o rozměru n zabere n + 32 taktů, přičemž na použitém hardwaru jeden takt trvá 10 ns; Konstanta 32 souvisí s použitím děličky s vysokou propustností, která však má dlouhou latenci (díky tomu je pak konstanta u lineární složky časové složitosti nízká). Počet taktů nutný pro výpočet  $\chi^2$  odpovídá počtu detektorů.

V rámci práce [43] byly ve VHDL implementovány dvě metody pro řešení soustavy lineárních rovnic: blokově paralelizovaná Gaussova – Jordanova eliminace bez pivotace a řádkově paralelizovaná Gaussova – Jordanova eliminace s řádkovou pivotací. Podle očekávání je blokově paralelizovaná varianta velmi rychlá. Řešení soustavy o n rovnicích jí v ideálním případě (silně regulární matice, což matice regularizované úlohy je) trvá 11 n + 5 taktů. To naznačuje, že v předchozích podkapitolách uvažovaná rekonstrukce o 438 pixelech by s touto implementací mohla splnit deadline 50 µs. Bohužel má metoda dvě podstatné vady: vysokou prostorovou složitost a numerickou nestabilitu. Kvůli prostorové složitosti je design na použitém hardwaru omezen na řešení soustavy o 25 rovnicích. Z kvadratické závislosti lze odhadnout, že pro řešení soustavy o 438 rovnicích by byl třeba hypotetický čip obsahující třistakrát více logických prvků. Závažná numerická nestabilita přitom byla pozorována už u soustav, které se do čipu vejdou.

Řádkově paralelizovaná Gaussova – Jordanova eliminace řeší soustavu o n rovnicích v čase  $n^2 + 18 n + 6$  taktů [43]. S lehkým využitím pipeliningu bylo dosaženo vysoké propustnosti té části obvodu, která provádí sérii řádkových úprav. Zároveň s řádkovými úpravami probíhá také výběr pivotu pro další sérii úprav. Díky těmto optimalizacím má kvadratická část časové složitosti nejnižší možnou váhu; I tak je však metoda pomalejší než ta předchozí. V limitu 50 µs zvládne vyřešit soustavu odpovídající rekonstrukci o přibližně šedesáti pixelech. Omezení dané velikostí čipu je podobné, dané hlavně tím, že při této velikosti designu dojdou hardwarové násobičky. Spotřeba vyhledávacích tabulek tentokrát není kritická, což je dáno

nejen efektivním využitím násobiček, ale také blokových pamětí, které předchozí metoda využít neumí (každý ze 116 paměťových bloků může v jednom taktu přečíst jedno osmnáctibitové slovo a jedno zapsat, což je pro blokově paralelizovanou metodu nedostatečné).

Popsané dvě přímé metody byly doplněny o paralelní implementaci Gaussovy – Seidelovy metody. Ta je tentokrát řešena jako modul provádějící jeden iterační krok metody, a organizace těchto kroků a přepočtů matice úlohy je přenechána řídícímu modulu. Výpočet jednoho iteračního kroku s přepočtem matice trvá 4 n + 38 taktů, bez přepočtu pak 3 n + 2 takty; Další redukci konstanty u lineární složky časové složitosti zvolená implementace metody neumožňuje kvůli závislostem dat (viz kód 2.2). Předpokládáme-li, že pro dostatečně kvalitní rekonstrukci jednoho snímku je třeba šest iterací, z toho tři s přepočtem matice, pak se do obvyklého deadlinu vejde výpočet rekonstrukce o rozlišení přibližně 230 pixelů. Požadavek na kvalitu je ovšem odhadnut z výsledků z předchozí podkapitoly, a v praxi se může lišit (přímé metody takovou variabilitu nemají). Na testovacím čipu bohužel zmíněného rozlišení nelze dosáhnout, protože design potřebuje na každý sloupec řešené soustavy jeden paměťový blok a jednu násobičku. Použití hardwarových násobiček tedy omezuje velikost soustavy na 58 vhodně zkombinujeme hardwarové násobičky rovnic, a pokud S násobičkami implementovanými pomocí vyhledávacích tabulek, lze maximální velikost soustavy odhadnout přibližně na 100 rovnic. Řešení takovéto úlohy pak lze získat v deadlinu 25 µs, použijeme-li výše zmíněnou kombinaci iteračních kroků a přepočtů matice, případně lze v deadlinu 50 µs vyřešit úlohu přesněji.

Simulovat je možné i design, který se do skutečného zařízení nevejde, například proto, že zabírá více vstupních a výstupních pinů, než kolik je na čipu k dispozici. Pro případné budoucí využití byla ovšem prozkoumána i možnost snížit počet vstupů pomocí metod feature selection. Byla použita metoda plus-*r*-minus-*l* s r = 2 a l = 1; Tyto poměrně nízké konstanty umožňují navrhnout vhodné kombinace detektorů od jednoho po všech fungujících 68 i v případě, že pro každou zkoumanou kombinaci je třeba provést několik stovek rekonstrukcí o rozlišení 438 pixelů. Metoda plus-*r*-minus-*l* byla vyzkoušena pro optimalizaci odchylky polohy i relativní normy rozdílu řešení (obojí od referenční rekonstrukce se všemi dostupnými detektory). Druhé z těchto kritérií bylo úspěšnější, soudě podle toho, že s ním ve většině případů bylo dosaženo nejen nižších hodnot normy rozdílu řešení, ale i nižších odchylek polohy. Výsledky optimalizované na sadě trénovacích dat jsou dobře reprodukovatelné na datech testovacích (alespoň při stejném rozlišení); V obou případech jde ovšem o experimentální data z omezeného rozsahu výbojů a hrozí, že po významnější změně parametrů plazmatu přestanou navržené kombinace detektorů spolehlivě fungovat. Získaný vývoj stochastické odchylky polohy a normy rozdílu profilů s přibývajícími detektory je znázorněn na obrázku 3.12. Obrázek 3.13 ukazuje navrženou kombinaci 16, respektive 32 detektorů. V těchto kombinacích lze vysledovat tendenci k rovnoměrnému rozložení chord, ale ne zcela důslednou; Zejména s ní nesouhlasí výběr sedmi sousedících chord v případě s 32 detektory. Bohužel není jasné, jestli jde o overtraining nerozpoznaný pomocí testovacích dat, nebo skutečně o uspořádání blízké optimálnímu.



Obr. 3.12: Závislost relativní normy rozdílu optimalizované a referenční rekonstrukce (vlevo) a střední stochastické odchylky polohy od referenční rekonstrukce (vpravo) na počtu použitých datových kanálů. K určení závislostí byla použita experimentální data odlišná od těch, na kterých probíhala optimalizace.



Obr. 3.13: Rozložení vybraných chord na průřezu komorou COMPASSu (vlevo) a totéž po Radonově transformaci (černé hvězdičky odpovídají chordám, šedá plocha představuje obraz vnitřku komory). Znázorněny jsou nejlepší nalezené kombinace 32 (nahoře) a 16 chord.

Na závěr bylo provedeno srovnání simulovaných výsledků FPGA s referenčními algoritmy. Byla použita paralelní implementace Gaussovy – Seidelovy metody provádějící přepočet matice úlohy po každých dvou iteracích, to celé třikrát pro každý snímek (numerická stabilita řádkově paralelizované Gaussovy – Jordanovy eliminace byla demonstrována již dříve [43]). Pro účely simulace se počítalo s rozlišením 104 pixelů. Pro srovnání byla použita stejná metoda implementovaná v MATLABu (hlavní rozdíl spočívá v přesnější reprezentaci čísel), a dále referenční algoritmus používající přímou metodu k řešení soustavy lineárních rovnic a rozlišení 438 pixelů. Algoritmy s iterační metodou (v MATLABu i VHDL) byly vyzkoušeny se vstupem ze všech 68 dostupných detektorů a se vstupem z 32 detektorů vybraných metodou plus-r-minus-l, aby se poznalo, zda má větší vliv omezení vstupních dat nebo reprezentace čísel. Jako testovací data posloužilo 300 snímků vybraných z výbojů #8713, #8766 a #8780. Toto omezené testování souvisí s faktem, že běh simulace je o šest řádů pomalejší než odpovídající fungování skutečného obvodu (simulováno na dvoujádrovém procesoru Intel® Celeron<sup>®</sup> CPU N2840 taktovaném na 2.16 GHz s 4 GB operační paměti, pomocí programu ISim od firmy Xilinx); Rekonstrukce jednoho snímku tak trvá přibližně minutu. Lze uvažovat o urychlení simulace jejím počítáním s nižším časovým rozlišením (výchozí rozlišení je 1 ps), ale i tak patrně zůstane významně pomalejší než skutečné FPGA.

Má-li Gaussova – Seidelova metoda implementovaná ve VHDL k dispozici všechna data, liší se její výstupy od stejné metody v MATLABu o 3.0 %, měřeno relativní normou rozdílu. To je menší odchylka, než jakou způsobí omezení vstupních dat na 32 nejvíce vypovídajících detektorů (ta při daném rozlišení a na daných datech činí 6.4 %). Výstup metody omezené jak v reprezentaci čísel, tak v dostupných detektorech, se pak od té původní liší o 10.1 %. Obrázek 3.14 ukazuje porovnání rekonstrukcí získaných referenčním algoritmem a algoritmy s Gaussovou – Seidelovou metodou běžícími v MATLABu a v simulátoru FPGA. Pokud jde o střední stochastické odchylky v poloze, omezení reprezentace čísel má opět menší vliv (2.8 mm) než omezení počtu detektorů (5.7 mm). Od referenčního algoritmu s 438 pixely a mnoha iteracemi se přitom Gaussova – Seidelova metoda se 104 pixely v MATLABu liší jen o 2.4 mm, ta v FPGA pak o 4.0 mm. Stále není zřejmé, zda je takovéto měření dostatečně přesné pro řízení polohy plazmatu. Pokud se ukáže, že rozlišení 104 pixelů dostatečné není, existuje prostor pro zlepšení použitím většího čipu, například takového, do něhož se vejde zmíněný design pro 230 pixelů (implementace má lineární prostorovou složitost). Další zlepšení je možné pouze s použitím hardwaru pracujícího s vyšší frekvencí hodinového signálu; Například blokové paměti čipů z řady Virtex 7 od firmy Xilinx pracují s frekvencí až 600 MHz, což představuje šestinásobné zlepšení oproti testovanému hardwaru [79]. Při návrhu obvodu pro tyto frekvence je však třeba dbát na timing, což by patrně vyžadovalo několik zásahů do stávajícího designu.



Obr. 3.14: Porovnání referenční rekonstrukce o 438 pixelech (vlevo), rekonstrukce o 104 pixelech získané pomocí algoritmu s klouzavou iterací a Gaussovou – Seidelovou metodou implementovaného v MATLABu (uprostřed) a analogická rekonstrukce získaná ze simulace FPGA (vpravo)

# Závěr

Na tokamaku COMPASS je k dispozici bolometrická a měkká rentgenová diagnostika v uspořádání, které dovoluje tomograficky rekonstruovat emisivitu v daném oboru na průřezu plazmatem. Tomografie pracující s těmito diagnostikami je rutinně využívána při analýze experimentálních dat. Je založena na přímé inverzi projekční matice doplněné o regularizaci Tichonovova typu. Od svého vzniku byl algoritmus postupně optimalizován pro co nejvyšší rychlost a zároveň co nejvyšší věrohodnost výsledků. Naproti tomu tato práce se zabývá radikální optimalizací výpočetního času za cenu co nejmenší ztráty přesnosti některých aspektů výsledku. Cílem takové optimalizace je umožnit využití tomografie v rychlé zpětné vazbě na tokamaku COMPASS, jejíž smyčka probíhá s periodou 50 µs. Kritickým úkolem této smyčky je řídit vertikální polohu plazmatu, na což se také soustředí většina provedených testů.

V této a předchozí práci byla navržena řada možností, jak zkrátit trvání tomografického výpočtu. K těm jednodušším patří snížení rozlišení rekonstrukce a tím i dimenze řešené úlohy, případně snížení počtu iterací používaných při řešení nelineární úlohy. Iteraci lze dále urychlit dobrou volbou počátečního přiblížení řešení. Byla zkoumána i možnost rozšířit použití iteračních metod a urychlit je inicializací výsledkem z předchozího časového kroku. V některých případech je možné provést část výpočtu předem a tím zkrátit druhou část výpočtu, která probíhá mezi vstupem dat a výstupem výsledku. Konečně se dá využít dobrých možností paralelizace řady z používaných algoritmů a implementovat je na hardwaru, který takovou paralelizaci umožňuje. Z navržených optimalizací jich bylo několik implementováno a otestováno; Při testování byl kladen důraz na změření zhoršení kvality výstupů algoritmu, k němuž při řadě úprav dochází, a odhadnutí nejvyšší přípustné míry zjednodušení, aby algoritmus zůstal použitelný pro řízení polohy plazmatu.

Výstup tomografických algoritmů má podobu dvojrozměrného obrazu (vyvíjejícího se v čase). Pro extrakci informace o poloze profilu bylo navrženo několik metod využívajících hledání maxima, těžiště, případně fitování. Tyto metody byly otestovány s využitím dat modelových i naměřených skutečnou měkkou rentgenovou diagnostikou na tokamaku COMPASS. V řadě případů se vskutku ukazuje, že některé z těchto metod jsou vhodnější než jiné. Například výpočet těžiště celého profilu příliš úspěšný není, zřejmě kvůli své citlivosti na artefakty a okrajové rysy rekonstrukcí, které jsou pro určení polohy málo podstatné. Lepší výsledky podává například výpočet těžiště nejjasnějších 25 % plochy profilu. U všech metod byla pozorována systematická odchylka od dostupných referenčních hodnot; Ta je patrně způsobena nejen algoritmem přepočítávajícím profil na polohu, ale i chybami v tomografii, případně (u dat z reálných experimentů) nekompatibilitou mezi tomograficky a magneticky měřenými veličinami. V dalších testech se za měřítko úspěšnosti algoritmů považuje velikost té složky odchylky polohy, která se nedá odstranit afinní transformací.

Na modelových i experimentálních datech byly ověřeny možnosti snižování rozlišení a počtu iterací. Ukazuje se, že u většiny metod regularizace lze bezpečně snížit rozlišení na přibližně 200 pixelů, přičemž lze určit vertikální polohu profilu emisivity s odchylkou v řádu jednotek milimetrů. Některé výsledky naznačují, že podobně velké odchylky se dá dosáhnout i s nižším rozlišením, ovšem zdá se, že tento závěr silně závisí na okolnostech (výsledky jsou jiné pro reálná a modelová data, pro některá data funguje konkrétní druh regularizace nebo konkrétní hodnota rozlišení). Proto byly algoritmy osvědčující se v rekonstrukci používaných testovacích dat vyzkoušeny i na jiné, do jisté míry nezávislé sadě dat, a v některých případech byl skutečně odhalen overtraining. Algoritmy využívající Fisherovy informace se v provedených testech zdají být poměrně robustní, ovšem vzhledem k pozorované variabilitě odchylek vznikajících ve většině tomografických metod se zdá být vhodné zopakovat testy po každé změně algoritmu či uspořádání experimentu.

Pro přiměřeně přesné určení polohy (tak, aby odchylka zavedená optimalizací byla významně menší než odchylka ve výsledcích už přítomná) je nutné provést přibližně deset iterací referenčního tomografického algoritmu. S použitím inicializace předchozím výsledkem se dá tato potřeba omezit na jednu až několik iterací. Pak se ovšem ukazuje, že odchylka polohy není mezi snímky rozložena rovnoměrně, nýbrž je vyšší po rychlé změně tvaru profilu. Byl navržen filtr, který tento jev omezí za cenu zvýšení náhodného šumu v poloze; Není však jasné, zda je tato oprava pro řízení polohy nutná, respektive užitečná.

Dalšího urychlení výpočtu bylo dosaženo s použitím iteračních metod pro řešení soustavy lineárních rovnic, která se v úloze objevuje. Tyto metody lze opět inicializovat posledním dostupným výstupem tomografické rekonstrukce, což vede k omezení potřebných iterací na pět až deset (při rozlišení ve stovkách pixelů). Překvapivě dobře se přitom osvědčuje jednoduchá Gaussova – Seidelova metoda, ačkoli má tu nevýhodu, že nízkofrekvenční složky řešení v ní konvergují pomaleji než vysokofrekvenční. Mezi její výhody naopak patří snadné přizpůsobení pro řídké matice a (spolu s ostatními iteračními metodami) variabilita spočívající v možnosti vyvážení četnosti jejích iterací se vzorkovací frekvencí vstupních i výstupních dat a iteracemi metody používané k regularizaci.

Gaussova – Seidelova metoda byla implementována v jazyce VHDL a spolu s dříve implementovanou Gaussovou – Jordanovou eliminací byla ověřena její funkčnost v malém programovatelném hradlovém poli. Ukazuje se, že přímou metodou lze v daném časovém limitu vypočítat tomografickou rekonstrukci o rozlišení kolem 60 pixelů, u Gaussovy – Seidelovy metody je to přibližně 100 pixelů. Tato rozlišení jsou stále poměrně malá vzhledem k výše zmíněným výsledkům, ovšem v tomto ohledu lze dosáhnout zlepšení použitím výkonnějšího hradlového pole (zejména u Gaussovy – Seidelovy metody, která je v současné implementaci omezována spíše množstvím dostupných logických prvků než časovým limitem).

Výsledky dosahované s navrženými implementacemi naznačují, že na moderním hardwaru lze provádět tomografii měkkého rentgenového záření dostatečně rychle pro řízení polohy plazmatu na tokamaku COMPASS. Je ovšem zřejmé, že v případě potřeby lze design přizpůsobit i pro jiné účely, s jinými požadavky na rychlost, přesnost a spolehlivost. V první řadě lze uvažovat o řízení polohy plazmatu ve větších tokamacích, jejichž vertikální nestabilita může být řádově pomalejší, a lze tedy použít vyšší rozlišení, případně více iterací tomografických metod. Vzhledem k tomu, že všechny navržené algoritmy produkují kompletní tomografické rekonstrukce, mohou být použity i v jiných zpětnovazebních smyčkách, například pro řízení celkového fúzního výkonu měřeného z tomografie neutronů nebo hlídání obsahu nečistot v plazmatu; Tyto jevy jsou opět pomalejší než vertikální nestabilita, ovšem lze předpokládat, že na implementaci tomografie budou klást nové požadavky.

## Reference

- R. Pánek et al. *Reinstallation of the COMPASS-D Tokamak in IPP ASCR*. Czechoslovak Journal of Physics 56 (2006) B125-B137
- [2] M. Hron et al. Overview of the COMPASS CODAC system. Fusion Engineering and Design 89 (2014) 177-185
- [3] F. Janky et al. *Upgrade of the COMPASS tokamak real-time control system*. Fusion Engineering and Design 89 (2014) 186-194
- [4] R. Beňo, J. John. *Modelování zpětnovazebního řízení polohy plazmatu v tokamaku COMPASS*. Československý časopis pro fyziku 59 (2009) 242 245
- [5] F. Janky et al. Determination of the plasma position for its real-time control in the COMPASS tokamak. Fusion Engineering and Design 86 (2011) 1120-1124
- [6] L. C. Ingesson et al. *Tomography Diagnostics: Bolometry and Soft X-Ray*. Fusion Science and Technology 53 (2008) 528-576
- [7] D. Marocco et al. Combined unfolding and spatial inversion of neutron camera measurements for ion temperature profile determination in ITER. Nuclear Fusion 51 (2011) 053011
- [8] J. Mlynář et al. Soft X-ray tomography in support of impurity control in tokamaks.
   International Workshop & Summer School on Plasma Physics, Kiten, Bulgaria (2013)
- [9] D. Vezinet et al. *Impurity density derivation from bandpass soft x-ray tomography: applicability, perspectives and limitations*. Nuclear Fusion 54 (2014) 083011
- [10] M. Imríšek et al. Use of soft x-ray diagnostic on the COMPASS tokamak for investigations of sawteeth crash neighborhood and of plasma position using fast inversion methods. Review of Scientific Instruments 85 (2014) 11E433
- [11] F. H. Séguin et al. *Radiation-hardened x-ray imaging for burning-plasma tokamaks*. Review of Scientific Instruments 68 (1997) 753-756
- [12] Wikipedia contributors. Single-photon emission computed tomography.
   Wikipedia, The Free Encyclopedia
   <en.wikipedia.org/wiki/Single-photon\_emission\_computed\_tomography> (15. 11. 2015)
- [13] T. Tanimoto, T. Lay. Mantle dynamics and seismic tomography.
   Proceedings of the National Academy of Sciences of the United States of America 97 (2000) 12409-12410
- [14] Wikipedia contributors. *Muon tomography*. Wikipedia, The Free Encyclopedia <en.wikipedia.org/wiki/Muon\_tomography> (15. 11. 2015)
- [15] R. A. Mintzer et al. Computed vs. conventional tomography in evaluation of primary and secondary pulmonary neoplasms. Radiology 132 (1979) 653-659

- [16] A. C. Kak, M. Slaney. Principles of Computerized Tomographic Imaging. Society for Industrial and Applied Mathematics (2001)
- [17] F. Natterer. Numerical methods in tomography. Acta Numerica (1999) 107-141
- [18] J. Sunnegårdh. Iterative Filtered Backprojection Methods for Helical Cone-Beam CT. Linköping University, disertační práce (2009)
- [19] X. Pan et al. Why do commercial CT scanners still employ traditional, filtered back-projection for image reconstruction? Inverse Problems 25 (2009) 1230009
- [20] Wikipedia contributors. *Allan McLeod Cormack*. Wikipedia, The Free Encyclopedia <en.wikipedia.org/wiki/Allan\_McLeod\_Cormack> (15. 11. 2015)
- [21] A. M. Cormack. Representation of a Function by Its Line Integrals, with Some Radiological Applications. Journal of Applied Physics 34 (1963) 24-29
- [22] L. C. Ingesson. Advanced analysis methods for SXR diagnostics Tomography. Fusion for Energy, prezentace (2015)
- [23] Wikipedia contributors. *Gaussian elimination*. Wikipedia, The Free Encyclopedia <en.wikipedia.org/wiki/Gaussian\_elimination> (15. 11. 2015)
- [24] A. H. Andersen, A. C. Kak. Simultaneous Algebraic Reconstruction Technique (SART): a Superior Implementation of the ART Algorithm. Ultrasonic Imaging 6 (1984) 81-94
- [25] E. F. Oliveira et al. *Comparison among Tomographic Reconstruction Algorithms with a Limited Data*. International Nuclear Atlantic Conference, Belo Horizonte, Brazil (2011)
- [26] Y. Kaganovsky et al. Compressed samplig strategies for tomography. Journal of the Optical Society of America A 31 (2014) 1369-1394
- [27] Z. Zhao et al. Noise, sampling, and the number of projections in cone-beam CT with a flat-panel detector. Medical Physics 41 (2014) 061909
- [28] Wikipedia contributors. *Tikhonov regularization*. Wikipedia, The Free Encyclopedia <en.wikipedia.org/wiki/Tikhonov\_regularization> (15. 11. 2015)
- [29] Y. Lu et al. Selective-diffusion regularization for enhancement of microcalcifications in digital breast tomosynthesis reconstruction. Medical Physics 37 (2010) 6003-6014
- [30] P. C. Hansen. *The truncated SVD as a method for regularization*. Stanford University, Numerical Analysis Project (1986) 534-553
- [31] Wikipedia contributors. *Fisher information*. Wikipedia, The Free Encyclopedia <en.wikipedia.org/wiki/Fisher\_information> (15. 11. 2015)
- [32] M. Anton et al. X-ray tomography on the TCV tokamak.Plasma Physics and Controlled Fusion 38 (1996) 1849-1878
- [33] D. Mazon. Soft x-ray tomography for real-time application: present status at Tore Supra and possible future developments. Review of Scientific Instruments 83 (2012) 063505

- [34] M. Nair et al. *Morozov's discrepancy principle under general source conditions*. Zeitschrift für Analysis und Anwendungen. 22 (2003) 199-214
- [35] V. Löffelmann. Tomografie měkkého rentgenového záření pro řízení tokamaku v reálném čase. FJFI ČVUT, bakalářská práce (2013)
- [36] A. Wingen et al. *Regularization of soft-X-ray imaging in the DIII-D tokamak*. Journal of Computational Physics 289 (2015) 83-95
- [37] M. Odstrčil. *Tomografie plazmatu na tokamaku COMPASS*. FJFI ČVUT, bakalářská práce (2010)
- [38] J. Mlynář et al. Tikhonov regularisation adapted to the real-time tomography.
   1st TM IAEA on Fusion Data Processing, Validation and Analysis, Nice, France (2015)
- [39] G. De Tomassi et al. *Current, position, and shape control in tokamaks*. Fusion Science and Technology 59 (2011) 486-498
- [40] Y. Gribov et al. Plasma vertical stabilisation in ITER. Nuclear Fusion 55 (2015) 073021
- [41] C. Angioni et al. *Tungsten transport in JET H-mode plasmas in hybrid scenario;* experimental observations and modelling. Nuclear Fusion 54 (2014) 083028
- [42] J. Mlynář et al. Focus on: JET, The European Centre of Fusion Research. EFDA, Culham Science Centre, EFDA-JET-R(07)01 (2007)
- [43] V. Löffelmann. Začleňování analýzy měkkého rentgenového záření do systému zpětnovazebního řízení tokamaku COMPASS. FJFI ČVUT, výzkumný úkol (2014)
- [44] R. Woods. FPGA-Based Implementation of Signal Processing Systems. Wiley (2008). ISBN 978-0-470-03009-7
- [45] C. Cullinan et al. Computing Performance Benchmarks among CPU, GPU and FPGA. Worcester Polytechnic Institute, E-project (2012)
- [46] I. Kuon, J. Rose. Measuring the Gap Between FPGAs and ASICs.
   IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 26 (2007) 203-215
- [47] Xilinx. 7 Series FPGAs Overview; Product Specification (15. 11. 2015) <www.xilinx.com/support/documentation/data\_sheets/ds180\_7Series\_Overview.pdf>
- [48] M. Parker. Understanding Peak Floating-Point Performance Claims. Altera, Technical White Paper (2014)
- [49] J. Lamoureux, S. J. E. Wilton. FPGA Clock NetworkArchitecture: Flexibility vs. Area and Power. Proceedings of the ACM/SIGDA 14th International Symposium on FPGA (2006)
- [50] Wikipedia contributors. *VHDL*. Wikipedia, The Free Encyclopedia <en.wikipedia.org/wiki/VHDL> (21. 11. 2015)

- [51] Wikipedia contributors. *Verilog*. Wikipedia, The Free Encyclopedia <en.wikipedia.org/wiki/Verilog> (21. 11. 2015)
- [53] P. Greisen et al. Evaluation and FPGA Implementation of Sparse Linear Solvers for Video Processing Applications. Circuits and Systems for Video Technology, IEEE Transactions 23 (2013) 1402-1407
- [54] A. Bogdanov et al. SMITH A Parallel Hardware Architecture for fast Gaussian Elimination over GF(2). Workshop on Special-purpose Hardware for Attacking Cryptographic Systems, Conference Records (2006)
- [55] Y. Saad. Iterative Methods for Sparse Linear Systems. SIAM (2003). ISBN 978-0898715347
- [56] Wikipedia contributors. *Conjugate gradient method*. Wikipedia, The Free Encyclopedia <en.wikipedia.org/wiki/Conjugate\_gradient\_method> (15. 11. 2015)
- [57] R. Barrett et al. Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods. SIAM (1994). ISBN 978-0898713282
- [58] R. D. Falgout. *An Introduction to Algebraic Multigrid*. Lawrence Livermore National Laboratory (2006)
- [59] G. Peters, J. H. Wilkinson. On the Stability of Gauss-Jordan Elimination with Pivoting. Communications of the ACM 18 (1975) 20-246
- [60] Wikipedia contributors. *LU decomposition*. Wikipedia, The Free Encyclopedia <en.wikipedia.org/wiki/LU\_decomposition> (15. 11. 2015)
- [61] N. J. Higham. *How Accurate is Gaussian Elimination?* Cornel University, technical report (1989)
- [62] S. Donfack et al. On Algorithmic Variants of Parallel Gaussian Elimination: Comparison of Implementations in Terms of Performance and Numerical Properties. University of Tennessee, EECS Department, technical report (2013)
- [63] J. Goodman. Introduction of Pipelining Optimisations into Gaussian Elimination. Draft Proceedings of the 3rd Scottish Functional Programming Workshop (2001)
- [64] Wikipedia contributors. *Cholesky decomposition*. Wikipedia, The Free Encyclopedia <en.wikipedia.org/wiki/Cholesky\_decomposition> (15. 11. 2015)
- [65] J. D. Hogg. *High Performance Cholesky and Symmetric Indefinite Factorizations with Applications*. University of Edinburgh (2010)
- [66] D. Yang et al. *Performance Comparison of Cholesky Decomposition on GPUs and FPGAs*. University of Tennessee, technical note (2010)
- [67] S. Toledo. Computing the Cholesky Factorization of Sparse Matrices.
   Tel Aviv University. <www.tau.ac.il/~stoledo/Support/chapter-direct.pdf> (15. 11. 2015)
- [68] Y. Zhou. Fourier Analysis and Local Fourier Analysis for Multigrid Methods. Johannes Kepler Universität, Linz, diplomová práce (2009)
- [69] H. Courtecuisse, J. Allard. *Parallel Dense Gauss-Seidel Algorithm on Many-Core Processosrs*. High Performance Computation Conference, IEEE CS Press (2009)
- [70] D. Gordon. Parallel ART for image reconstruction in CT using processor arrays.
   The International Journal of Parallel, Emergent and Distributed Systems (2006) 365-380
- [71] H. Erdoğan, J. A. Fessler. Ordered subsets algorithms for transmission tomography. Physics in Medicine and Biology 44 (1999) 2835-2851
- [72] F. Xu et al. On the efficiency of Iterative Ordered Subset Reconstruction Algorithms for Acceleration on GPUs.
   Computer Methods and Programs in Biomedicine 98 (2010) 261-270
- [74] Xilinx. Spartan-6 Family Overview; Product Specification <www.xilinx.com/support/documentation/data\_sheets/ds160.pdf> (15. 11. 2015)
- [75] H. Motoda, H. Liu. Feature Selection, Extraction and Construction. Communication of IICM (Institute of Information and Computing Machinery, Taiwan) 5 (2002) 67-72
- [76] P. Somol et al. Adaptive floating search methods in feature selection. Pattern Recognition Letters 20 (1997) 1157-1163
- [77] MathWorks. Mldivide, \, Solve systems of linear equations Ax = b for x MATLAB <www.mathworks.com/help/matlab/ref/mldivide.html> (15. 11. 2015)
- [78] A. Forsgren. On the Behavior of the Conjugate-Gradient Method on Ill-Conditioned Problems. Royal Institute of Technology, Stockholm, technical report (2006)
- [79] Xilinx. Virtex-7 T and XT FPGAs Data Sheet: DC and AC Switching Characteristics; Product Specification. (28. 12. 2015) <www.xilinx.com/support/documentation/data\_sheets/ds183\_Virtex\_7\_Data\_Sheet.pdf>

## Příloha A: Obsah CD

text.pdf		elektronická verze této práce
MATLAB/		
trenovac	iData.mat	sada 3940 snímků měkké rentgeové diagnostiky používaná v této práci jako trénovací data
dets_tre	novaci.txt	seznam detektorů používaných při rekonstrukci trénovacích dat
testovac	iData.mat	sada 3000 snímků používaná jako testovací data
dets_tes	tovaci.txt	seznam detektorů používaných při rekonstrukci testovacích dat
tomograf	ie_dvacykly.m	skript provádějící tomografifkou rekonstrukci regularizovanou Fisherovou informací s parametry optimalizovanými ve dvou vnořených iteračních cyklech
tomograf	ie_jedencyklus.m	skript provádějící tomografifkou rekonstrukci s parametry optimalizovanými v jednom cyklu
tomograf	ie_jedencyklus_GSVD.m	skript provádějící tomografifkou rekonstrukci s parametry optimalizovanými v jednom cyklu s využitím metody GSVD
tomograf	ie_zadnycyklus.m	skript provádějící tomografifkou rekonstrukci s klouzavou iterací
tomograf	ie_zadnycyklus_GS.m	skript provádějící tomografifkou rekonstrukci s využitím Gaussovy – Seidelovy metody
getZ_CoM	. m	funkce počítající vertikální polohu plazmatu jako těžiště části profilu emisivity
getZ_fit	. m	funkce počítající vertikální polohu plazmatu pomocí fitování Gaussovy funkce
getZ_max	. m	funkce počítající polohu plazmatu jako polohu maxima shlazeného profilu emisivity
plusK_mi	nusL.m	skript provádějící feature selection
saveBinD	ata.m	skript převádějící data z MATLABu do podoby čitelné pro simulátor FPGA
loadBinD	ata.m	skript převádějící výstup simulace do MATLABu

## Python/

	matice/	složka obsahuje data popisující projekční matici a matici hodnotícího funkcionálu
	getChi2.py	skript generující soubor getChi2.vhdl (python getChi2.py T_*.txt)
	getRadekMatice_GS.py	<pre>skript generující soubor getRadekMatice.vhdl (python getRadekMatice_GS.py T_*.txt TT_*.txt H_koef_*.txt)</pre>
	redukce.py	skript generující část souboru tomography_hardwaretest.vhdl (python redukce.py)
	tomography_test.py	skript generující soubor tomography_test.vhdl (python tomography_test.py T_*.txt)
VHDL/		složka obsahuje kompletní projekt prostředí Xilinx ISE implementující regularizovanou tomografickou rekonstrukci