FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

Zadání diplomové práce

Název:	Integrace metody ITO do nástroje ParaCell
Student:	Bc. Michal Čermák
Vedoucí:	doc. Ing. Ivan Šimeček, Ph.D.
Studijní program:	Informatika
Obor / specializace:	Teoretická informatika
Katedra:	Katedra teoretické informatiky
Platnost zadání:	do konce letního semestru 2021/2022

Pokyny pro vypracování

1) Seznamte se základními pojmy a fyzikálními modely v oblasti krystalických látek. Zaměřte se na vlastnosti rentgenového záření ve spojení s metodou práškové difrakce používané pro získání strukturních parametrů zkoumané krystalické látky [1].

2) Nastudujte tzv. přímé metody pro řešení krystalové struktury využívající globální optimalizaci strukturního modelu a kompletního difrakčního obrazu.
3) Nastudujte vnitřní architekturu nástroje ParaCell [2], jehož autorem je vedoucí práce.

4) Navrhněte a implementujte vícevláknový algoritmus pro extrakci reálných mřížkových parametrů z naměřených hodnot obsažených v difrakčním záznamu pomocí metody ITO [3]. Diskutujte vhodnost použití různých datových struktur.

5) Algoritmus otestujte na klastru STAR pro data z volně dostupných vzorků experimentálně naměřených práškovou difrakcí a vyhodnoťte škálovatelnost implementovaného algoritmu.

.

[1] M. Rulf:Algoritmy výpočetní krystalografie DP ČVUT FIT,2014

[2] ParaCell home page https://sourceforge.net/projects/paracell/

[3] J. W. Visser: "A fully automatic program for finding the unit cell from powder data", Journal of Applied Crystallography,vol.2,no.3,p. 89-95, 1969.

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Diplomová práce

Integrace metody ITO do nástroje ParaCell

Bc. Michal Čermák

Katedra Teoretické Informatiky
Vedoucí práce: doc. Ing. Ivan Šimeček, Ph.D.

5. května 2022

Poděkování

Děkuji svému vedoucímu za vstřícný přístup a pomoc při psaní této práce. Dále děkuji své rodině za podporu během studia a psaní této práce, především své mámě.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo"), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu) licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 5. května 2022 .....

Odkaz na tuto práci

Čermák, Michal. *Integrace metody ITO do nástroje ParaCell.* Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2022.

Abstrakt

Tato práce se zabývá zpracováním dat získaných pomocí metody práškové difrakce. Konkrétně se zabývá jednou z metod indexace těchto dat, ITO. Práce obsahuje základní pojmy z oblasti krystalografie a s ní spjatou metodou práškové difrakce. Dále obsahuje popis metody ITO a také její implementace do programu ParaCell a výsledky jejího testování.

Klíčová slova Krystalografie, Prášková difrakce, OpenMP

Abstract

The main topic of this thesis is the processing of data obtained through powder diffraction. In particular, this thesis focuses on the ITO method. This thesis contains basic crystallographic terminology and that of a related method called powder diffraction. Furthermore, it contains a description of the ITO method, its implementation in the ParaCell program and test results of said implementation.

Keywords Crystallography, Powder diffraction, OpenMP

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Lidé se odjakživa snaží porozumět světu, ve kterém žijí. Mezi to patří i např. určování vlastností látek a mezi ty patří mj. i jejich struktura. Tato práce se zabývá strukturou krystalických látek, konkrétně parametry jejich krystalových mřížek. Zkoumá tak pravidelnost a periodicitu, která se přirozeně vysktyuje téměř všude v přírodě a zejména právě v krystalických látkách.

K určování parametrů krystalových mřížek využívá dat získaných pomocí metody zvané prášková difrakce. Ty obsahují záznamy o úhlech, pod kterými se od nějaké krystalické látky ve formě prášku odrazily nejintenzivnější paprsky rentgenového záření (odtud prášková difrakce). Z těchto úhlů jsou pak pomocí různých metod vypočítány parametry krystalické látky, jejíž vzorek byl podroben právě práškové difrakci.

Tato práce se zaměřuje na jednu z metod indexace těchto dat, konkrétně metodu ITO. Metoda ITO byla navržena J. W. Visserem v roce 1969 v jeho článku. Cílem je integrace této metody do programu ParaCell. Tento program slouží právě k indexaci dat z práškové difrakce a tedy k určování parametrů krystalových mřížek daných látek. K tomuto účelu je v ParaCellu již implementováno několik metod.

Kapitola 1

Základní pojmy a cíle práce

Tato kapitola obsahuje základní pojmy z krystalografie a další pojmy užívané v této práci.

1.1 Cíle práce

Hlavním cílem této práce, jak už její název napovídá, je implementace metody ITO do programu ParaCell, což je program pro indexaci krystalických látek z dat získaných pomocí práškové difrakce.

1.2 Skupenství látek

Obvykle rozlišujeme 3 základní látková skupenství:

- pevné
- kapalné
- plynné

Někdy se jako 4. druh skupenství uvádí plazma.

1.2.1 Pevné látky

Informace o pevných látkách jsou čerpány z [6]. Pevné látky rozdělujeme na amorfní a krystalické. Rozdílem mezi těmito dvěma typy je pravidelnost uspořádání částic látky.

U amorfních látek lze pozorovat pravidelnost přibližne do 10 nm.

Krystalické látky dále dělíme na *monokrystaly* a *polykrystaly*. V monokrystalu se uspořádání částic periodicky opakuje. Polykrystal se skládá z mnoha krystalů (též zrn) s rozměry od 10 µm do několika mm. Jednotlivá zrna mají periodickou struktura, jsou vůči sobě však náhodně orientována.

1.3 Krystalografie

Krystalografie je věda, která se zabývá pevnými krystalickými látkami a jejich strukturou. Informace v této sekci jsou čerpány z [7] a [1].

1.3.1 Krystal

Krystal je periodicky se opakující struktura částic. Při abstrakci se používá tzv. *ideální krystal*, který je nekonečný a jeho struktura se tedy periodicky opakuje do nekonečna. Reálné krystaly však nejsou ideální krystaly, můžou se lišit např. z těchto důvodů:

- Jejich struktura je konečná, tedy ohraničená, což narušuje periodicitu.
- V některých částech krystalu může docházet k nahrazení atomy jiných prvků.
- Během vzniku krystalu mohou vznikat různé poruchy v jeho krystalové struktuře.

1.3.2 Krystalová mřížka

Krystalová mřížka je způsob abstrakce periodicity částic v rámci ideálního krystalu. Je charakterizována 6 parametry $a, b, c, \alpha, \beta, \gamma$, kde a, b, c jsou délky vektorů $\vec{a}, \vec{b}, \vec{c}$ a α, β, γ úhly svírané mezi nimy (α označuje úhel mezi $\vec{b}$ a $\vec{c}$ atd.). Periodicita je tak vyjádřena jako přičítání celočíselných násobků vektorů $\vec{a}, \vec{b}, \vec{c}$ k počátku. Body, které lze takto vyjádřit ($i\vec{a} + j\vec{b} + k\vec{c}; i, j, k \in Z$), označíme mřížové body.

Úhly α, β, γ jsou tradičně uváděny ve stupních a parametry a, b, c v jednotkách Å (Ångström, $1\,\text{Å} = 0{,}1\,\text{nm}$).

1.3.2.1 Základní buňka

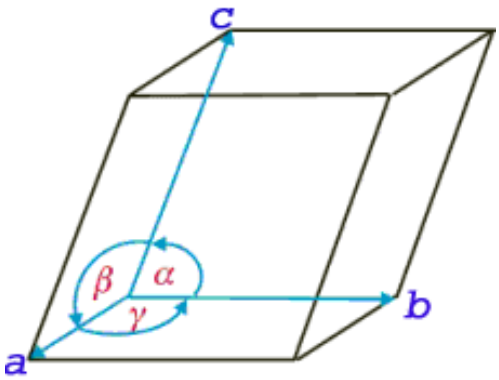
Jako základní buňka se označuje rovnoběžnostěn vyhraničený 8 sousedními mřížovými body. Parametry daného rovnoběžnostěnu odpovídají mřížovým parametrům, tedy má délky stran a, b, c a úhly svírané mezi nimy α, β, γ. V ideálním krystalu je díky dokonalé periodicitě obsah každé základní buňky identický a liší se tedy pouze pozicí v prostoru. Objem základní buňky značíme V a myslíme jím objem daného rovnoběžnostěnu. Obrázek 1.1 ilustruje příklad základní buňky.

1.3.2.2 Bravaisovy typy mřížek

Typicky dělíme mřížky na 14 typů, označované Bravaisovy typy. Pro účely této práce je sloučíme do těchto 7 základních typů mřížek:

- triklinická: $a \neq b \neq c$

Obrázek 1.1: Příklad základní buňky (převzaté z [1])



- monoklinická: $a \neq b \neq c, \alpha = \gamma, \beta > 90°$
- ortorombická: $a \neq b \neq c, \alpha = \beta = \gamma = 90°$
- tetragonální: $a = b \neq c, \alpha = \beta = \gamma = 90°$
- hexagonální: $a = b \neq c, \alpha = \beta = 90°, \gamma = 120°$
- romboedrická: $a = b = c, \alpha = \beta = \gamma \neq 90°$
- kubická: $a = b = c, \alpha = \beta = \gamma = 90°$

1.3.3 Mřížová rovina a osnova rovin

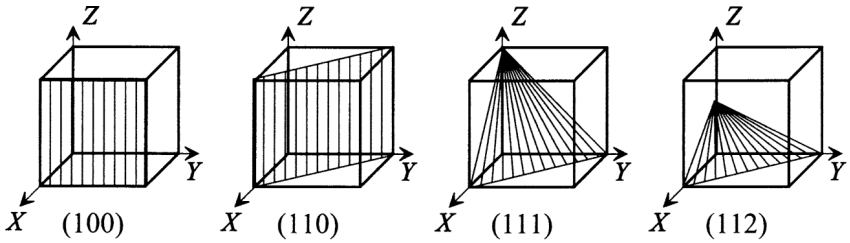
Mřížová rovina je rovina, která je jednoznačně dána 3 různými body z krystalové mřížky a reprezentuje tak řez napříč krystalem. Množiny mřížových rovin, které jsou navzájem rovnoběžné, a pro každé 2 sousední roviny v nich platí, že mají mezi sebou stejnou vzdálenost d, označujeme jako *osnovu rovin*.

K označení osnov rovin se používají Millerovy indexy, tj. 3 celá čísla označovaná jako h, k a l. Osnova rovin hkl obsahuje mřížové roviny, které jsou rovnoběžné s rovinou, která protíná 3 body určené vektory $\frac{\vec{a}}{h}, \frac{\vec{b}}{k}$ a $\frac{\vec{c}}{l}$. Pokud je 1 nebo 2 z hodnot h, k, l rovno 0, pak jsou roviny rovnoběžné s odpovídajícími vektory $\vec{a}, \vec{b}, \vec{c}$ resp. Ukázky těchto rovin jsou pro ilustraci na obrázku 1.2.

1.3.4 Reciproká mřížka

Reciproká mřížka se zavádí kvůli zjednodušení některých výpočtů a interpretaci některých experimentů. Reciprokou mřížku tvoří body, které leží na normálových vektorech osnov rovin hkl ve vzdálenosti $\frac{1}{d_{hkl}}$ (inverzní hodnota vzdálenosti d mezi sousedními rovinami v osnově hkl). Každý bod tak reprezentuje vlastnosti jednotlivých osnov rovin, konkrétně jejich orientaci a vzdálenost d_{hkl}.

Obrázek 1.2: Příklad rovin hkl (převzaté z [2])



Reciproká mřížka je charakterizována obdobně jako krystalová mřížka 6 parametry označovanými $a^*, b^*, c^*, \alpha^*, \beta^*, \gamma^*$, kde a^*, b^*, c^* jsou délky vektorů $\vec{a^*}, \vec{b^*}, \vec{c^*}$ a $\alpha^*, \beta^*, \gamma^*$ úhly svírané mezi nimy. Platí následující vztahy:

$$\vec{a^*} = \frac{\vec{b} \times \vec{c}}{V}, a^* = \frac{1}{d_{100}} = \frac{bc \sin \alpha}{V}, \cos \alpha^* = \frac{\cos \beta \cos \gamma - \cos \alpha}{\sin \beta \sin \gamma}$$

$$\vec{b^*} = \frac{\vec{a} \times \vec{c}}{V}, b^* = \frac{1}{d_{010}} = \frac{ac \sin \beta}{V}, \cos \beta^* = \frac{\cos \alpha \cos \gamma - \cos \beta}{\sin \alpha \sin \gamma}$$

$$\vec{c^*} = \frac{\vec{a} \times \vec{b}}{V}, c^* = \frac{1}{d_{001}} = \frac{ab \sin \gamma}{V}, \cos \gamma^* = \frac{\cos \alpha \cos \beta - \cos \gamma}{\sin \alpha \sin \beta}$$

Dále platí:

$$\vec{a}\vec{b^*} = \vec{a}\vec{c^*} = \vec{b}\vec{a^*} = \vec{b}\vec{c^*} = \vec{c}\vec{a^*} = \vec{c}\vec{b^*} = 0$$

tedy vektory $\vec{b}$ a $\vec{c}$ jsou kolmé na vektor $\vec{a^*}$ atd. To je celkem zřejmé vzhledem k tomu, že rovina procházející osami $\vec{b}$ a $\vec{c}$ je rovnoběžná s rovinami z osnovy s hkl indexy 100. Jejich normálový vektor je tedy kolmý na oba tyto vektory.

Jednotlivé body reciproké mřížky lze pak vyjádřit jako součet $i\vec{a^*} + j\vec{b^*} + k\vec{c^*}$ kde $i, j, k \in Z$

Pro jednotlivé typy mřížek lze pak vyjádřit $\frac{1}{d_{hkl}^2}$ následovně pomocí hkl a jejich přímých parametrů:

- triklinická:

$$\left(\frac{h^2}{a^2} \sin^2 \alpha + \frac{k^2}{b^2} \sin^2 \beta + \frac{l^2}{c^2} \sin^2 \gamma + \frac{2kl}{bc}(\cos \beta \cos \gamma - \cos \alpha) + \right.$$
$$\left. + \frac{2lh}{ca}(\cos \gamma \cos \alpha - \cos \beta) + \frac{2hk}{ab}(\cos \alpha \cos \beta - \cos \gamma)\right.$$

- monoklinická:

$$\frac{h^2}{a^2 \sin^2 \beta} + \frac{k^2}{b^2} + \frac{l^2}{c^2 \sin^2 \beta} - \frac{2hl \cos \beta}{ac \sin^2 \beta}$$

- ortorombická:

$$\frac{h^2}{a^2} + \frac{k^2}{b^2} + \frac{l^2}{c^2}$$

- tetragonální:
$$\frac{h^2+k^2}{a^2}+\frac{l^2}{c^2}$$
- hexagonální:
$$\frac{4}{3a^2}(h^2+k^2+hk)+\frac{l^2}{c^2}$$
- romboedrická:
$$\frac{1}{a^2}(\frac{(h^2+k^2+l^2)\sin^2\alpha+2(hk+kl+lh)(\cos^2\alpha-\cos\alpha)}{1+2cos^3\alpha-3\cos^2\alpha})$$
- kubická:
$$(h^2+k^2+l^2)\frac{1}{a^2}$$

1.4 Prášková difrakce

Informace v této sekci jsou čerpány z [7] a [1].

Prášková difrakce je metoda používaná pro určení parametrů krystalové mřížky nějakého reálného krystalu. Při této metodě je používána krystalická látka v práškové formě, efektivně tedy ve formě mnoha velmi malých krystalů, které jsou různě orientovány (polykrystal). Na tento prášek je vysíláno záření s dostatečně krátkou vlnovou délkou (např. nejčastěji je to rentgenové záření) a je sledováno a zaznamenáno, pod jakými úhly dopadá nejintenzivnější záření. V takovém případě totiž dochází ke konstruktivní interferenci (tj. počátky vln odražených paprsků jsou od sebe vzdáleny nějaký násobek jejich vlnové délky a mají stejný směr, což vede ke zvýšení amplitudy těchto vln) 2 nebo více odrážených paprsků.

Při použití krystalické látky ve formě prášku je tedy vzhledem k velkému počtu jednotlivých krystalů velmi pravděpodobné, že bude obsahovat krystaly správně orientované pro pozorování tohoto jevu.

1.4.1 Rentgenové záření

Podle [8] rentgenové záření je elektromagnetické záření s vlnovou délkou od 10^{-12}m do 10^{-8}m. Vzniká při dopadu rychle se pohybujících elektronů na povrch kovové elektrody. Rentgenové záření bylo objeveno v roce 1895 W. C. Röntgenem, když studoval výboje v plynech.

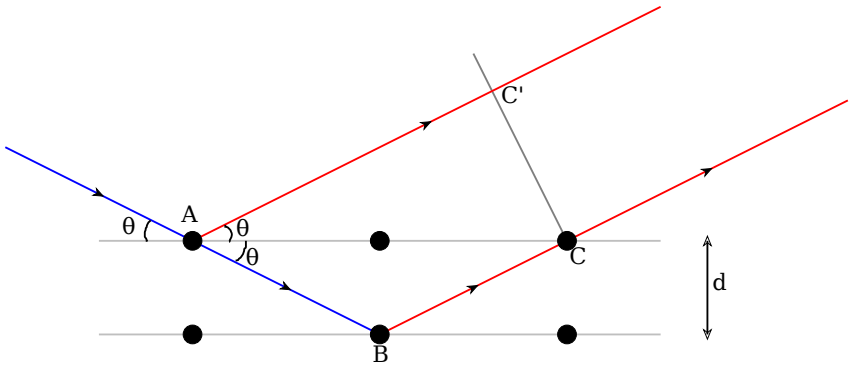
Podle [9] při práškové difrakci je nejčastěji používaná měděná elektroda, která vytváří paprsky s vlnovou délkou 1,5418 Å.

1.4.2 Braggův zákon

Braggův zákon popisuje podmínky, kdy dochází ke konstruktivní interferenci při odrazech rentgenového záření v krystalu.

1.4.2.1 Odvození braggovy rovnice

Obrázek 1.3: Odrážení paprsku (převzaté z [3])



Na obrázku 1.3 lze vidět odražené paprsky. Podle bodů A, B, C, C' na obrázku odvodíme, za jakých podmínek dochází ke konstruktivní interferenci. Ta nastane, když oba paprsky urazí stejnou vzdálenost. Tedy pokud:

$$AB + BC - AC' = n\lambda$$

kde n je nějaké kladné celé číslo a λ je vlnová délka paprsku. Tedy pokud je rozdíl vzdálenosti paprsků rovný nějakému násobku vlnové délky. Lze odvodit následující:

$$AB = BC = \frac{d}{\sin\theta}$$

$$AC = \frac{2d}{\operatorname{tg}\theta}$$

Z toho plyne:

$$AC' = AC\cos\theta = \frac{2d}{\sin\theta}\cos^2\theta$$

Tedy nakonec:

$$n\lambda = \frac{d}{\sin\theta} - \frac{2d}{\sin\theta}\cos^2\theta = \frac{d}{\sin\theta}(1-\cos^2\theta) = 2d\sin\theta$$

Tím jsme dospěli k braggově rovnici:

$$n\lambda = 2d\sin\theta \tag{1.1}$$

V rovnici 1.1 $n\lambda$ označuje nějaký celý násobek vlnové délky použitého paprsku, d označuje vzdálenost mezi rovinami odrazu hkl a θ je úhel odrazu. Z této rovnice se dále odvozuje a označuje hodnota Q:

$$Q = \frac{1}{d^2} = (\frac{2\sin\theta}{\lambda})^2 \tag{1.2}$$

Proměnnou n lze bez újmy na obecnosti z rovnice vypustit.

Q hodnoty jsou důležité kvůli možnosti jednoduchého výpočtu z parametrů reciproké mřížky z následující rovnice:

$$Q_{hkl} = \frac{1}{d_{hkl}^2} = h^2A + k^2B + l^2C + klD + hlE + hkF \tag{1.3}$$

kde:

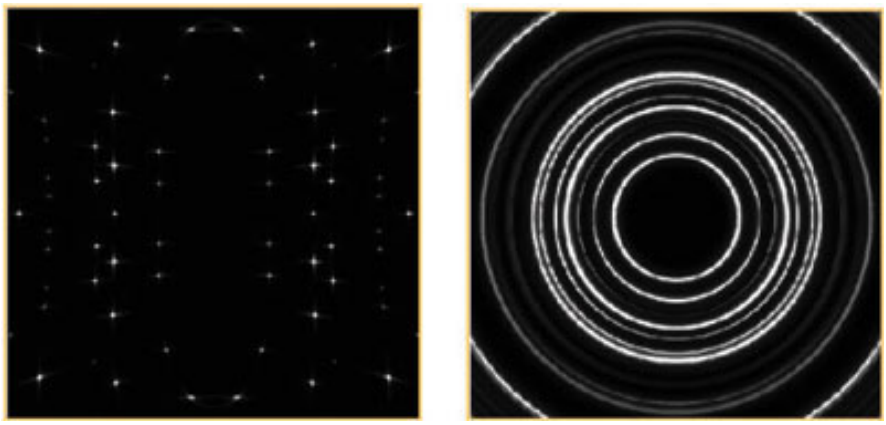
$$A = a^{*2}, B = b^{*2}, C = c^{*2}, D = b^*c^* \cos\alpha^*, E = a^*c^* \cos\beta^*, F = a^*b^* \cos\gamma^*$$

1.4.3 Difrakční záznam

Difrakční záznam reprezentuje výsledky provedené práškové difrakce. Zpravidla obsahuje úhly 2θ, pro které dosáhla intenzita dopadlého rentgenového záření nějaké požadované minimální hodnoty, a příslušné intenzity. Typicky záznam obsahuje okolo 20 hodnot 2θ. Tyto jednotlivé hodnoty pak obvykle označujeme jako *reflexe*.

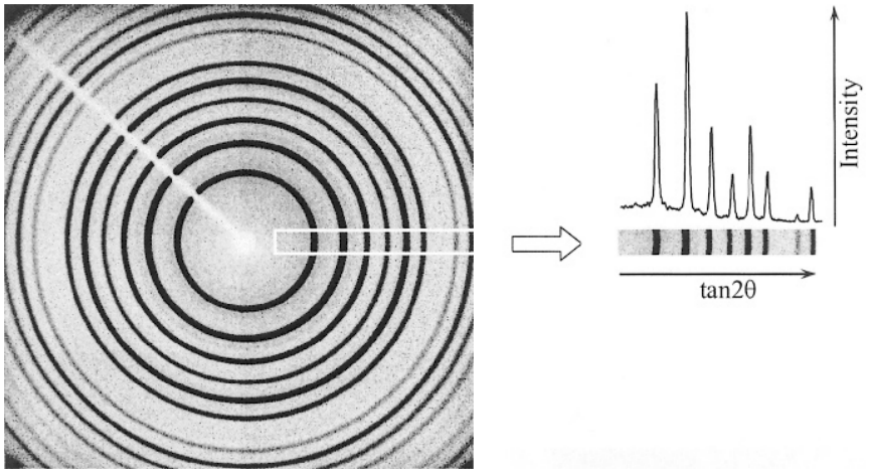
Na obrázku 1.4 lze vidět obraz difrakce. Levý obrázek ilustruje difrakci pomocí monokrystalu. Pravý obrázek pak ilustruje práškovou difrakci. V monokrystalu dochází difrakcí k bodovému vzoru. V polykrystalu jsou pak tyto body rozprostřeny do kruhů.

Obrázek 1.4: Příklad difrakce (převzaté z [4])

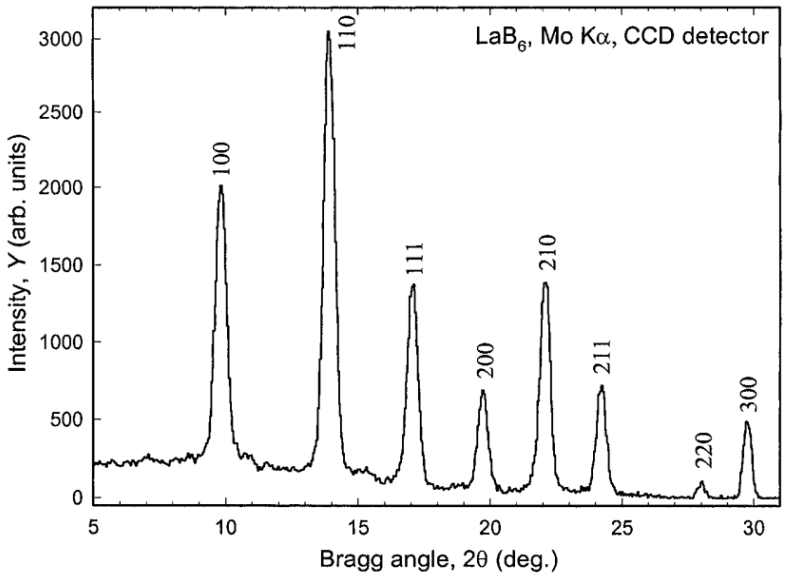


Z těchto obrazů se pak získávají hodnoty intenzit způsobem ilustrovaným obrázky 1.5. Vybere se část difrakčního obrazu a podle ní se vypočtou intenzity v příslušných místech $\mathrm{tg}\,2\theta$, které lze pak vynést do grafu jako je ilustrováno na obrázku 1.6. Z tohoto grafu lze pak vyčíst hodnoty 2θ, kde dochází k nejvyšším relativním intenzitám. Takto získané hodnoty se pak stanou součástí difrakčního záznamu.

Obrázek 1.5: Převod difrakčního obrazu na intenzity (převzaté z [5])



Obrázek 1.6: Příklad difrakce (převzaté z [5])



1.4.4 Indexace

Indexací difrakčního záznamu se rozumí výpočet parametrů základní buňky a tedy krystalové mříže nějaké krystalické látky z dat jejího difrakčního záznamu. K tomuto účelu bylo vyvinuto několik metod. Mezi tyto metody patří např. Treor, Dicvol, McMaille a také metoda ITO, na kterou se zaměřuje tato práce. Metoda ITO je podrobně popsána v následující kapitole.

Při indexaci hledáme pro jednotlivé typy mřížek následující přímé para-

metry (nebo jim odpovídající reciproké parametry):

- triklinická: $a, b, c, \alpha, \beta, \gamma$
- monoklinická: a, b, c, β
- ortorombická: a, b, c
- tetragonální: a, c
- hexagonální: a, c
- romboedrická: a, α
- kubická: a

1.4.4.1 Metoda TREOR

Metoda Treor je popsána v [10]. Zjednodušeně metoda postupuje následovně. Systematicky přiřazuje určitému počtu (odpovídající počtu hledaných parametrů pro daný typ mřížky) reflexí v záznamu různé kombinace hkl indexů z předem stanovené množiny. Tím po dosazení do rovnice 1.3 získá soustavy lineráních rovnic, jejichž řešení vede k získání kandidátů na parametry krystalové mřížky.

Tedy máme rovnice ve tvaru:

$$Q_{hkl} = h^2x_1 + k^2x_2 + l^2x_3 + hkx_4 + hlx_5 + klx_6$$

Tím získáme matici M (vyplněná indexy u neznámých $x_1 \dots x_6$). Dále máme vektor neznámých $\vec{X} = (x_1 \dots x_6)^{\mathrm{T}}$ a vektor příslušejících Q hodnot $\vec{Q} = (Q_1 \dots Q_6)^{\mathrm{T}}$, kde $Q_1 \dots Q_6$ jsou vybrané reflexe (typicky z nejmenších hodnot 2θ v záznamu). Řešíme tedy rovnici:

$$M\vec{X} = \vec{Q}$$

Řešením je tedy:

$$\vec{X} = M^{-1}\vec{Q}$$

Dosazováním různých kombinací hkl indexů z předem určené množiny a vybraných Q hodnot do této rovnice pak získáme kandidáty na řešení.

1.4.4.2 Metoda DICVOL

Metoda Dicvol podle [11]. Její princip spočívá v práci s intervaly možných hodnot pro jednotlivé parametry krystalové mřížky. Pokud interval mřížky z daného intervalu dokáží vysvětlit všechny reflexe, rozdělí se interval na dva (dichotomie), jinak se zahodí. Takto se postupuje rekurzivně dále s novou množinou intervalů.

Tedy např. můžeme řešit kubickou soustavu v přímých parametrech mřížky. Máme interval hodnot parametru a, označme krajní hodnoty tohoto intervalu $a_{\min}, a_{\max}$. Pomocí rovnice pro výpočet získáme krajní hodnoty Q, které interval pokrývá.

$$Q_1 = \frac{h^2 + k^2 + l^2}{a_{\min}^2}$$

$$Q_2 = \frac{h^2 + k^2 + l^2}{a_{\max}^2}$$

Do rovnice dosazujeme přípustné parametry hkl a ukládáme nejmenší a největší nalezené Q hodnoty jako $Q_{\min}$ a $Q_{\max}$ resp. Pokud pro všechny Q hodnoty v záznamu platí, že:

$$Q_{\min} - \epsilon < Q < Q_{\max} + \epsilon$$

poté prohlásíme interval za přípustný a dále ho dělíme. V opačném případě interval zahodíme.

Podobně můžeme pracovat i v prostoru reciprokých parametrů mřížky.

1.4.4.3 Grid Search

Dále je také možné problém řešit také např. pomocí systematického prohledávání parametrů mřížky (přímých nebo i reciprokých parametrů), tzv. Grid search. To se realizuje tak, že pro každý z parametrů se určí nějaký interval a krok mezi hodnotami v tomto intervalu. Řešení se pak hledá mezi všemi kombinacemi možných hodnot parametrů, které jsou voleny z daných intervalů, v rozmezí daného kroku.

1.4.4.4 de Wolffovo kritérium M_{20}

K vyhodnocení kvality potenciálních řešení lze použít různá kritéria. Jedním z nich je např. de Wolffovo kritérium popsané v [12], označované M_{20}.

$$M_{20} = \frac{Q_{20}}{|\Delta 2\theta| N_{20}} \tag{1.4}$$

Q_{20} označuje Q hodnotu 20. reflexe v daném difrakčním záznamu, N_{20} označuje počet vypočtených teoretických reflexí odpovídajících naměřeným až do Q_{20} a $|\Delta 2\theta|$ je průměrný rozdíl mezi naměřenými a vypočitanými reflexemi.

Podle de Wolffa se hodnoty M_{20} pro kvalitní difrakční záznamy pohybují mezi hodnotami 20 až 60. Dále pokud je neoindexovaných reflexí nejvýše 2 a hodnota M_{20} je větší než 10, je rozumné považovat řešení za možné správné. Hodnoty menší než 6 jsou dobrým důvodem k pochybování o správnosti řešení.

1.4.4.5 F_N index

Druhým kritériem pro ohodnocení kvality řešení je Smithův a Snyderův F_N index popsaný v [13]. Počítá se následujícím způsobem:

$$F_N = \frac{N}{|\Delta 2\theta| N_c} \tag{1.5}$$

kde N je počet naměřených reflexí, N_c je počet možných vypočítaných reflexí až do hodnoty reflexe s indexem N a $|\Delta 2\theta|$ je průměrný rozdíl mezi naměřenými a vypočítanými teoretickými hodnotami reflexí.

Podle Smitha a Snydera je F_N index vhodnější k řazení vypočítaných řešení od nejlepšího k nejhoršímu, než M_{20}.

1.5 Paralelní zrychlení

Jako zrychlení $S(n)$ v kontextu paralelního výpočtu chápeme poměr doby výpočtu s použití jednoho vlákna ku době paralelního výpočtu. Počítáme tedy takto:

$$S(n) = \frac{T_1}{T_n}$$

kde T_1 je doba výpočtu jednoho vlákna a T_n je doba výpočtu při použití n vláken. Pokud platí, že $S(n) = \Theta(n)$ pak mluvíme o lineárním zrychlení.

1.6 Metoda nejmenších čtverců

Informace o této metodě jsou čerpány z [14]. Metoda nejmenších čtverců je metoda aproximace řešení soustav rovnic, které mají více rovnic než neznámých. Pro účely této práce se zaměříme na lineární metodu nejmenších čtverců. Máme tedy soustavu lineárních rovnic:

$$X\vec{\beta} = \vec{y}$$

Naším cílem je potom minimalizovat kvadratickou chybu (odtud název metoda nejmenších čtverců):

$$||X\vec{\beta} - \vec{y}||^2$$

Metoda lineárních nejmenších čtverců má přesné analytické řešení $\vec{\beta'}$ (tedy za předpokladu, že matice $X^{\mathrm{T}}X$ má inverzi) a tím je:

$$\vec{\beta'} = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}\vec{y}$$

Kapitola 2

Metoda ITO

Tato práce vychází z metody ITO popsané v [15]. Následující kapitola obsahuje volný překlad z tohoto článku. Metoda vychází z rovnice 1.3.

Základem metody je uvažování osnovy rovin hkl, kde jedno z h, k, nebo l je rovné 0. Rovnice 1.3 se pak redukuje na:

$$Q_{m,n} = m^2Q' + n^2Q'' + mnR \tag{2.1}$$

kde m a n může odpovídat libovolné dvojici z h, k, l a Q', Q'', R pak odpovídajícím proměnným $A \dots F$.

Úpravou rovnice 2.1 získáme rovnost pro R:

$$R = (Q_{m,n} - m^2Q' - n^2Q'')/mn \tag{2.2}$$

2.1 Hledání zón

Za Q' a Q'' postupně dosazujeme do rovnice 2.2 všechny možné naměřené Q hodnoty. Pro každou z těchto dvojic nalezneme potenciální hodnoty R následujícím způsobem. Pro několik kladných celých hodnot m a n dosadíme do rovnice za $Q_{m,n}$ naměřené Q hodnoty a uložíme si absolutní hodnotu $|R|$ (stačí uvažovat pouze absolutní hodnotu, jelikož m nebo n může být i záporné). Hodnoty $|R|$, které se pro danou dvojici vyskytují opakovaně, uložíme společně s danou dvojicí Q' a Q'' jako jednu z možných zón. Uvažujeme pouze hodnoty, které se opakují alespoň r-krát, kde r je nějaká konstanta.

Vzhledem k nepřesnostem při měření Q hodnot nelze předpokládat, že spočítané $|R|$ hodnoty si budou přesně rovny, proto je potřeba přidat nějakou toleranci. Původní článek navrhuje např. vynásobit $|R|$ nějakou kladnou konstantou, zaokrouhlit na nejbližší celé číslo a poté uvažovat hodnoty za rovné, pokud rozdíl těchto hodnot je menší než nějaká další konstanta.

2.2 Úprava nalezených zón

Takto nalezené zóny, tedy trojice Q', Q'' a R, jsou následně upraveny podle následujících pravidel:

- Pokud $Q' = Q''$:

$$Q' = \frac{2Q' - R}{4}, Q'' = \frac{2Q'' + R}{4}, R = 0$$

- Pokud $R = Q'$:

$$Q' = \frac{Q' + R}{8}, Q'' = Q'' - \frac{Q' + R}{8}, R = 0$$

- Pokud $R = Q''$:

$$Q' = Q' - \frac{Q'' + R}{8}, Q'' = \frac{Q'' + R}{8}, R = 0$$

- Pokud $R > Q'$:

$$Q' = Q', Q'' = Q' + Q'' - R, R = 2Q' - R$$

Parametry upravených zón jsou nadále vylepšeny pomocí metody nejmenších čtverců. Tedy pro hodnoty m a n, pro které je vypočtená hodnota $Q_{m,n}$ v toleranci nějaké naměřené hodnoty Q, minimalizujeme chybu $(Q_{m,n} - Q)^2$.

Dále je pro všechny zóny vypočtena pravděpodobnost, že byly nalezeny náhodou, která je použita pro kvalitativní ohodnocení dané zóny. Označíme tuto pravděpodobnost C a počítáme následujícím způsobem:

$$C = \frac{N_c!}{N_o!(N_c - N_o)!} p^{N_o} (1 - p)^{N_c - N_o} \tag{2.3}$$

kde N_o je počet dvojic m a n, pro které vypočtená hodnota $Q_{m,n}$ je v toleranci nějaké naměřené hodnoty Q, a N_c je celkový počet testovaných dvojic m a n. Dále $p = \frac{N2\epsilon}{Q_{max}}$, kde N je počet naměřených Q hodnot a ϵ je tolerance kdy naměřené a vypočítané Q hodnoty považujeme za identické. Inverzní hodnotu této pravděpodobnosti $\frac{1}{C}$ označíme jako kvalitu dané zóny.

2.3 Kombinace zón

Pro všechny dvojice zón, které dosahují nějaké požadované podmínky na jejich kvalitu $\frac{1}{C}$, se pokusíme najít jejich průnik a tedy parametry $A \ldots F$. To je provedeno tak, že pro každou zónu jsou spočítány 4 nejmenší Q hodnoty,

konkrétně $Q', Q'', Q' + Q'' - R, Q' + Q'' + R$. Tyto hodnoty jsou porovnány. Pokud je nalezena společná Q hodnota, lze definovat průnik těchto zón jako:

$$A = Q_c, B = Q_{1,1}, C = Q_{2,1}, E = Q_c + Q_{2,1} - Q_{2,2}, F = Q_c + Q_{1,1} - Q_{1,2}$$

kde Q_c je nalezená společná hodnota, $Q_{1,1}, Q_{2,1}$ jsou nejmenší Q hodnoty 1. a 2. zóny různé od Q_c, obdobně $Q_{1,2}, Q_{2,2}$ jsou 2. nejmenší hodnoty různé od Q_c.

Dále je nutné dopočítat parametr D. Ten je spočítán obdobným způsobem jako R, ale z upravené rovnice 1.3:

$$D = (Q - h^2A - k^2B - l^2C - hlE - hkF)/(kl) \tag{2.4}$$

Za hodnoty h, k, l dosazujeme:

$$h \in \langle -2, 2\rangle, k \in \{-2, -1, 1, 2\}, l \in \{1, 2\}$$

Jako vhodné kandidáty pro hodnotu D uvažujeme určitý počet nejvíce se opakujících hodnot D při dosazení všech možných trojic h, k, l do rovnice 2.4.

Na závěr jsou provedeny tyto úpravy:

- Pokud $A = B$ a $F = 0$:

$$A' = \frac{A + B - F}{4}, B' = \frac{A + B + F}{4}, C' = C,$$

$$D' = \frac{D - E}{2}, E' = \frac{D + E}{2}, F' = 0$$

- Pokud $F = A$ nebo $F = B$:

$$A' = \frac{A}{4}, B' = B - \frac{A}{4}, C' = C, D' = D - \frac{E}{2}, E' = \frac{E}{2}, F' = 0$$

Obdobně pak pro kombinace A, C, E a B, C, D.

2.4 Návrh potenciálních vylepšení algoritmu

Byly navrženy celkem 4 relativně výrazné změny k možnému zlepšení kvality výstupu.

2.4.1 Více hodnot pro parametry Q' a Q''

Vylepšením oproti originálnímu algoritmu je, že za Q' a Q'' se při hledání parametrů Q', Q'' a R dosazují kromě naměřených Q hodnot také hodnoty $\frac{Q}{4}$, $\frac{Q}{9}$ a $\frac{Q}{16}$. To protože některé difkrakční záznamy neobsahují reflexe rovin hkl typu 001 atp. Mohou ale obsahovat např. reflexe 002, z tohoto důvodu má smysl uvažovat hodnoty $\frac{Q}{4}$ atd.

2.4.2 Změna výpočtu kvality zón

Dále je vzhledem k povaze ohodnocení zón možný výskyt zón se stejným ohodnocením, což vede ke kolizi při výběru nejlepších zón ke kombinaci. Návrhem je úprava funkce pro výpočet ohodnocení na jednu z těchto 3 možností:

$$\frac{1}{C_1} = \frac{1}{1 + \mathrm{e}^{-\frac{1}{\Delta Q}}} \frac{1}{C}$$

$$\frac{1}{C_2} = N_o + \frac{1}{1 + \mathrm{e}^{-\frac{1}{\Delta Q}}}$$

$$\frac{1}{C_3} = N_o - \frac{1}{|Q' + Q'' + R|}$$

kde ΔQ označuje průměrný rozdíl naměřených Q hodnot difrakčního záznamu od hodnot vypočtených pomocí parametrů dané zóny. Nápad za hodnotou $\frac{1}{C_1}$ je jednoduché přeškálování původní hodnoty tak, aby zóny, které více odpovídají záznamu měli lepší skóre. Druhá možnost je obdobná, základem je však namísto pravděpodobnosti nalezení zóny s daným počtem N_o vypočtených Q hodnot přímo jejich počet N_o. Obdobně ve třetí možnosti, ta však namísto nepřesnosti v rozdílu naměřených a vypočtených Q hodnot penalizuje malé hodnoty Q' a Q'' a R, které vedou na větší délky os krystalové mřížky resp. úhlů mezi nimi. Jinými slovy, při stejné kvalitě zón tak preferujeme ty, co vedou na menší objem výsledné základní buňky.

2.4.3 Jiné úpravy nalezených zón

Při úpravě zón se provádí změna pokud $A = B$ a F není rovné 0, tzn. pokud jsou 2 osy krystalové mříže stejně dlouhé a nejsou na sebe kolmé. Některé minerály však takové parametry mříže mají (viz. např. záznamy Kogarkoite_0000305 a Loweite_0000200 v databázi dostupné z [16]). Obdobně se provádí změna při finální úpravě parametrů celé krystalové mřížky pokud $A = B$ a F není rovné 0. Návrhem je tedy tyto změny neprovádět.

Dalším návrhem je opět změna při úpravě zón. V případě $Q > R$ bude provedena změna inspirovaná procesem kombinace zón. Obdobně jsou spočítány 4 nejmenší Q hodnoty $Q', Q'', Q' + Q'' - R, Q' + Q'' + R$. Označíme-li tyto hodnoty Q_1 až Q_4, potom:

$$Q' = Q_1, Q'' = Q_2, R = |Q_1 + Q_2 - Q_3|$$

2.4.4 Návrh pro případ méně kvalitních dat

Algoritmus velmi závisí na vstupních datech a jejich kvalitě. To by do určité míry mohlo být kompenzováno uvažováním hodnot $Q - \epsilon$ a $Q + \epsilon$, kde ϵ je nějaká konstanta, jako kandidáty pro Q' a Q'' při hledání zón. V případném spojení s návrhem 2.4.1 budeme uvažovat navíc následující hodnoty:

- $Q - \epsilon$, $Q + \epsilon$
- $\frac{Q-\epsilon}{4}$, $\frac{Q+\epsilon}{4}$

Použití vícero by příliš negativně ovlivnilo dobu výpočtu.

Kapitola 3

ParaCell a použité softwarové technologie

ParaCell je nástroj pro indexaci dat práškové difrakce, dostupný z [17]. Jeho hlavními autory jsou doc. Ing. Ivan Šimeček, Ph.D., vedoucí této práce a Ing. Jan Rohlíček, Ph.D., oponent práce. Je vyvíjen tak, aby byl multiplatformní, vícevláknový a kde je možné využíval technolgii CUDA pro výpočet na grafické kartě. Aktuální verze ParaCellu obsahuje algoritmy MGLS, DICVOL, TREOR a Grid search. Cílem této práce je rozšířit ParaCell o metodu ITO. Tyto metody, kromě MGLS, byly popsány v předchozích kapitolách.

3.1 Struktura ParaCellu

Implementaci ParaCellu lze rozdělit do 3 hlavních modulů. První modul se stará o čtení vstupních dat a konfiguračního souboru. Druhým jsou pak implementace různých metod, které ze vstupních dat vypočítají kandidáty krystalové mřížky odpovídající vstupním datům. Třetí modul pak tyto kandidáty vyhodnotí. Vyhodnocení může proběhnout pomocí CPU nebo GPU.

3.2 Vstup a výstup programu

Vstupem každého běhu programu je soubor s daty difrakčního záznamu a konfigurační soubor. Konfigurační soubor je textový soubor obsahující řádky ve tvaru parametr=hodnota. Lze pomocí něho např. zvolit algoritmus, který bude použit, nastavit počet vláken nebo parametry použitého algoritmu. Také je potřeba v konfiguračním souboru nastavit cílový typ hledané krystalové mřížky.

Výstupem programu je pak textový soubor obsahující výpis nalezených řešení, konkrétně parametry krystalové mřížky, objem základní buňky a hodnoty M_{20} a F_N indexů. Dále obsahuje pro určitý počet nejlepších řešení po-

drobný přehled obsahující *hkl* indexy, které odpovídají jednotlivým reflexím v záznamu, a odchylky vypočtených reflexí od naměřených.

3.3 OpenMP

Informace o OpenMP jsou čerpány z [18]. OpenMP umožňuje paralelizaci C, C++ kódu pomocí direktiv překladače a je použita v ParaCellu k jeho paralelizaci. OpenMP také umožňuje paralelizaci Fortran kódu.

V rámci implementace této metody byly využity tyto direktivy překladače OpenMP:

- `#pragma omp parallel` — Vytvoří tým vláken, které zpracují následující instrukci nebo blok kódu
- `firstprivate(seznam)` — Váže se k předchozí direktivě. V rámci paralelního bloku bude každé vlákno v týmu vlastnit svoji kopii proměnných v seznamu, inicializované na jejich hodnotu před vstupem do paralelního bloku.
- `#pragma omp for` — Způsobí, že iterace příslušného for cyklu budou provedeny paralelně vlákny v týmu.
- `schedule()` — Váže se k předchozí direktivě. Určuje způsob plánování iterací for cyklu na jedno z následujících:
  - `schedule(static)` — Každé vlákno má fixně přiřazené dané iterace cyklu.
  - `schedule(dynamic, N)` — Vláknům je za běhu přiřazováno N iterací, které ještě nebyly provedeny. Nová skupina iterací je vláknu přiřazena, jakmile svou původní dokončí.
  - `schedule(guided, N)` — Obdobně jako `schedule(dynamic, N)`, ale počet přiřazených iterací se postupně snižuje na N.
- `#pragma omp critical` — Takto označená instrukce nebo blok kódu bude v každém čase prováděna nanejvýš 1 vláknem (kritická sekce).

Mezi další důležité direktivy patří např.:

- `num_threads(n)` — Nastaví počet vláken, s kterými bude paralelní blok `#pragma omp parallel` zpracován na n.
- `private(seznam)` — V rámci daného paralelního bloku bude každé vlákno v týmu vlastnit svoji kopii proměnných v seznamu, neinicializovanou.
- `shared(seznam)` — V rámci paralelního bloku budou proměnné v seznamu sdílené mezi vlákny.

- `#pragma omp single` — Způsobí, že následující instrukce je zpracována pouze jedním vláknem.
- `#pragma omp master` — Způsobí, že následující instrukce je zpracována pouze hlavním vláknem (s indexem 0).
- `#pragma omp task` — Vytvoří úlohu, která je zpracována jedním z vláken v týmu
- `#pragma omp taskwait` — Počká na dokončení všech úloh v aktuálním paralelním bloku.

Dále OpenMP obsahuje několik knihovních funkcí. Jednou z nich je např. `omp_get_max_threads()`, která byla použita v této práci. Tato funkce vrací hodnotu maximálního a také vychozího počtu vláken, které OpenMP může využít v dané instanci běhu programu. Knihovní funkce lze v C++ použít díky hlavičkovému souboru `omp.h`. Mezi další příklady těchto funkcí patří:

- `omp_set_num_threads(n)` — Nastaví počet vláken na n pro následující bloky, které nespecifikují jinak pomocí `num_threads(n)`.
- `omp_get_num_threads()` — Funkce vrací hodnotu aktuálního počtu vláken v týmu.
- `omp_get_thread_num()` — Funkce vrací index volajícího vlákna v daném týmu.

3.4 Technologie CUDA

Technologie CUDA je podle [19] výpočetní platforma a programovací model vytvořený společností NVIDIA. Umožňuje implementaci paralelních výpočtů, které využívají NVIDIA GPU.

Kapitola 4

Implementace

Implementace byla provedena v jazyce C++, ve kterém je implementován nástroj ParaCell.

V rámci této práce byla pro program ParaCell implementována metoda ITO popsaná v kapitole 2. Implementace se skládá pouze ze samotné metody ITO, o ostatní procesy, jako je zpracování vstupu, vyhodnocení kandidátů na parametry mřížek a výstup, se starají ostatní, již implementované části ParaCellu. Implementaci metody rozdělíme implicitně rozdělíme do 3 hlavních částí:

- Nalezení zón (množina trojic parametrů Q', Q'' a R)
- Úprava nalezených zón
- Kombinace nalezených zón

4.1 Označení parametrů

V rámci implementace budeme používat několik důležitých konstant a nebo proměnných definujících chování algoritmu. Mezi ně patří:

- ϵ — Parametr používán k porovnávání Q hodnot a také reciprokých parametrů mřížky nebo zóny. Hodnoty Q_1 a Q_2 považujeme za shodné, pokud $|Q_1 - Q_2| < \epsilon$.
- $m_{\max}, n_{\max}$ — Při výpočtu R hodnot z rovnice 2.2 uvažujeme hodnoty m a n od 1 do $m_{\max}, n_{\max}$ resp.
- m_C, n_C — Při výpočtu kvality zóny uvažujeme její vypočítané reflexe z rovnice 2.1 pro hodnoty m od $-m_C$ do m_C a n od 0 do n_C, s výjimkou dvojic kde $m < 0$ a $n = 0$ (mají stejné hodnoty jako odpovídající $m > 0$ a $n = 0$).

- LS_{hkl} — Při úpravě mřížky metodou nejmenších čtverců uvažujeme její reflexe pro hodnoty h, k od $-LS_{hkl}$ do LS_{hkl} a l od 0 do LS_{hkl}. Obdobně při úpravě zón pomocí metody nejmenších čtverců uvažujeme její reflexe pro hodnoty m od $-LS_{hkl}$ do LS_{hkl} a n od 0 do LS_{hkl}. Opět vynecháváme opakující se hodnoty jako u výpočtu kvality zóny. Tedy pokud $l = 0$ a $k < 0$ nebo $k = 0$, $l = 0$ a $h < 1$. V případě zón pak vynecháváme hodnoty $m < 0$ a $n = 0$ stejně jako při výpočtu kvality zóny.

- R_{freq} — Při hledání R hodnot uvažujeme hodnoty, které se opakují alespoň R_{freq}-krát.

- Z_{top} — Pro kombinaci zón používáme Z_{top} zón s nejlepší kvalitou.

- D_{best} — Při hledání D hodnot vybereme D_{best} hodnot, které se opakovaly nejvícekrát.

V rámci implementace používáme Q hodnoty definované jako $Q = \frac{10^4}{d^2}$ pro konzistenci se značením v [15]. Tomu pak odpovídá i hodnota parametru ϵ.

4.2 Volba datových typů

K provádění většiny výpočtů s desetinnými čísly byl zvolen datový typ `float` s 32 bity přesnosti. Důvodem je rychlejší provedení výpočtů oproti typu `double`. Vzhledem k předpokládaným nepřesnostem při meření difrakčního zánamu není maximální přesnost výpočtu až tak důležitá. Zároveň je tento typ použit k výpočtům v rámci implementací většiny ostatních metod v ParaCellu.

Datový typ `double` s 64 bity přesnosti byl použit při výpočtu ohodnocení kvality zóny $\frac{1}{C}$, kde je přesnost namístě vzhledem ke komplikovanějšímu výpočtu s faktoriály.

Také byly použity speciálně datové typy pro ukládání zón (sestávající z 3 hodnot typu `float` pro parametry Q', Q'', R a hodnoty pro kvalitu zóny typu `double`) a ukládání D hodnot spolu s jejich frekvencí (1 `float` a 1 `int`).

4.3 Použité datové struktury

Jako jediná datová struktura byla použita struktura pole umožňující uložení více hodnot v paměti v řadě za sebou a umožňuje tak rychlý přístup k nim pomocí indexu, tj. vzdálenosti od pozice 1. hodnoty. Její použití se jevilo jako postačující pro všechny potřeby implementace algoritmu z hlediska časové i paměťové složitosti.

4.4 Hledání zón

Nejprve jsou pomocí for cyklů vypočteny R hodnoty pro jednotlivé dvojice Q hodnot a uložené do pole. Pole je následně seřazeno a následujícím způsobem jsou vybrány R hodnoty. Mějme indexy i, j v poli, označme $A[x]$ hodnotu x-tého prvku v poli a $A[x \dots y]$ průměrnou hodnotu prvků $x \dots y$. Postupujeme takto:

- Pokud $|A[j+1] - A[i \dots j]| < \epsilon$: $j = j + 1$
- Pokud $|A[j+1] - A[i \dots j]| \geq \epsilon$: Pokud od poslední nalezené zóny byl zvýšen index j, tak jsme našli novou zónu $Q', Q'', A[i \dots j]$ a postupujeme k její úpravě. Následně vždy $i = i + 1$.

4.5 Úprava zón

Nyní máme zónu Q', Q'', R, provedeme její úpravy podle sekce 2.2 nebo 2.4.3. Poté provedeme úpravu pomocí metody nejmenších čtverců tak, že minimalizujeme chybu $\sum(Q_{\text{calc}} - Q_{\text{obs}})^2$, kde Q_{calc} jsou hodnoty vypočtené z rovnice 2.1 s hodnotami m, n popsané v sekci 4.1 a $|Q_{\text{calc}} - Q_{\text{obs}}| < \epsilon$. V kontextu metody popsané v sekci 1.6 jsou řádky matice X tvořené hodnotami $m * m$, $n * n$ a $m * n$ a vektor $\vec{y}$ jsou hodnoty Q_{obs} odpovídající těmto hodnotám. Jednotlivé řádky jsou vážené hodnotou $\frac{1}{Q_{obs}}$ (vynásobené touto hodnotou). Efektivně tak minimalizujeme relativní chybu Q namísto absolutní. Po provedení metody nejmenších čtverců vypočteme kvalitu zóny $\frac{1}{C}$ s nově získanými parametry Q', Q'', R a do pole zón uložíme čtveřici $Q', Q'', R, \frac{1}{C}$.

Poté seřadíme pole se zónami podle jejich hodnot Q' a začneme odstraňovat duplikáty. To děláme tak, že vytvoříme nové pole, kam postupně vkládáme zóny z původního pole se zónami. Pokud v poli není zóna, pro kterou by platilo, že $|Q'_1 - Q'_2| < \epsilon$, $|Q''_1 - Q''_2| < \epsilon$ a $|R_1 - R_2| < \epsilon$, pak zónu do nového pole vložíme. Pokud to však pro některou zónu neplatí a tato zóna v novém poli má menší hodnotu $\frac{1}{C}$ (tedy menší kvalitu), než zóna, kterou se snažíme vložit, pak ji nahradíme touto zónou s lepší kvalitou. Parametry s indexy 1 a 2 označují parametry daných zón z původního a z nového pole.

4.6 Kombinace zón

Nové pole se zónami je seřazeno podle kvality a pro nejlepších Z_{top} je provedena kombinace podle sekce 2.3. Pro případné vzniklé parametry A, B, C, E, F je pak hledána hodnota D.

Výběr hodnot D je proveden téměř stejně jako v případě R hodnot s tím rozdílem, že výsledné D hodnoty jsou na závěr ukládány do pole spolu s počtem, kolikrát se vyskytovaly. Poté je z pole vybráno D_{best} hodnot s nejvyšší frekvencí, namísto přijmutí všech hodnot, které se vyskytují alespoň R_{freq}-krát.

Poté jsou opakovaně na parametry mřížky použity úpravy popsané v sekci 2.3, dokud dochází ke změně. Vypočítané parametry A, B, C, D, E, F jsou předány další části ParaCellu k vyhodnocení. Pokud je typ mřížky různý od triklinické a požaduje tedy méně než 6 parametrů, jsou použity vhodné permutace daných 6 parametrů (např. pro kubické postupně parametry A, B a poté C).

Finální optimalizace parametrů mřížky pomocí metody nejmenších čtverců byla implementována obdobně jako při úpravách zóny v sekci 4.5 s hodnotami hkl tak jak je popsáno v sekci 1.6. Nebyla však při měřeních použita, protože ParaCell obsahuje rychlejší verzi této metody paralelizovanou pomocí technologie CUDA.

4.7 Paralelizace

K paralelizaci byla využita technologie OpenMP, která je použita i v ostatních částech ParaCellu.

4.7.1 Paralelizace hledání zón

Paralelizace hlédání zón byla provedena paralelizací for cyklu, který určuje aktuální hodnoty Q' a Q'' při výpočtu R hodnot. Pro lepší paralelizaci byly jednotlivé cykly pro Q' a Q'' sloučeny v jeden. Parametr `schedule` byl zvolen `dynamic`, protože každá dvojice může mít různý počet validních R hodnot, což vede na nevyváženost výpočetní složitosti jednotlivých iterací.

4.7.2 Paralelizace úpravy zón

Paralelizace samotné úpravy zón je zahrnutá v paralelizaci for cyklu z předchozí sekce, protože následuje bezprostředně po nalezení R hodnoty. Datová závislost při ukládání zón do pole byla řešena pomocí kritické sekce.

Paralelizace odstranění duplikátních zón byla řešena následovně. Nejprve je seřazení zón provedeno jednoduchým paralelním mergesortem, který je součástí implementace. Pole se zónami je pak rovnoměrně rozděleno na tolik částí, kolik je vláken. Je vytvořen for cyklus s 1 iterací pro každou z těchto částí a ten je paralelizován pomocí `#pragma omp parallel for`. Parametr `schedule` byl zvolen `static`, protože na každé vlákno připadá jedna iterace. Datová závislost je řešena tak, že každé vlákno vkládá zóny do svého vlastního pomocného pole a poté je v kritické sekci přesune do společného pole.

Pilotní testy ukázaly, že mazání duplicitních zón je velmi důležité pro kvalitu výsledku. Protože každé vlákno vkládá neopakující se zóny do svého vlastního pomocného pole, je nutné zajistit, aby mezi těmito poli nebyly duplicitní záznamy. Toho je docíleno tak, že začátek každé z částí (a tedy i konec té předchozí) je posunut tak, aby jeho rozdíl hodnot Q' s předchozí zónou byl

větší než ϵ. To samozřejmě do velké míry může omezit paralelismus, ale díky vyloučení duplicitních zón mezi jednotlivými částmi zachovává kvalitu řešení.

4.7.3 Paralelizace kombinace zón

Zóny jsou seřazeny podle jejich kvality opět pomocí paralelního mergesortu. Poté je paralelizace provedena pomocí paralelizace for cyklu iterujícího přes index první zóny. Parametr `schedule` byl zvolen `dynamic`, protože výpočetní složitost se snižuje s vyšším indexem (vynecháváme duplicitní dvojice, kde index druhé zóny je menší než index první zóny). V této části implementace dochází k datovým závislostem při vkládání vygenerovaných řešení, což je řešeno v příslušné funkci ParaCellu pomocí kritické sekce.

Kapitola 5

Testování

5.1 Testovací platforma

Program byl testován na školním clusteru *star.fit.cvut.cz*. Jeho specifikace jsou:

- CPU: 2× Intel Xeon 2620 v2 — 12 jader celkem, 24 vláken (technologie Hyper-Threading)
- 32 GB RAM
- GPU: Tesla P100, GeForce RTX 2080 Ti, Tesla K40c, GeForce GTX 780 Ti a GeForce GTX 750

5.2 Testovací data

Program byl testován na difrakčních záznamech získaných z [16]. Konkrétně se jednalo o:

- 6 kubických
- 8 hexagonálních
- 25 monoklinických
- 17 ortorombických
- 6 tetragonálních
- 32 triklinických

Celkem tedy bylo 94 testovacích záznamů. Důvod distribuce počtu je z důvodu složitosti při určování daných mřížek (podle počtu parametrů). Tedy dat pro typ mřížek s více určovanými parametry je více.

5.3 Kritérium pro správnost řešení

Vypočtené řešení považujeme s velkou pravděpodobností za totožné s mřížkou uvedenou v záznamu z databáze, pokud jejich základní buňky mají stejný objem V, alespoň jeden z parametrů délek a, b nebo c a plochu S jedné ze stěn sousedící s danou stejně dlouhou hranou. Vzhledem k nepřesnostem, je třeba uvažovat určitý rozsah pro který prohlásíme dané hodnoty za stejné, ten je pro jednotlivé parametry zvolen následovně:

- $|a_1 - a_2| < 0{,}06\,\text{Å}$
- $|S_1 - S_2| < 0{,}5\,\text{Å}^2$
- $|V_1 - V_2| < 2\,\text{Å}^3$

Myšlenka za tímto kritériem je, že pokud je stejný objem V a stěny mají stejnou plochou S, označíme-li tyto stěny jako podstavu rovnoběžnostěnu, potom mají oba rovnoběžnostěny stejnou výšku. Jelikož je shodná délka jedné hrany podstavy (označme jako délku podstavy) a její obsah, je i šířka podstavy stejná.

Máme tedy 2 rovnoběžnostěny se stejnou výškou a také délkou a šířkou podstavy. Těmito 3 směry se tedy obě buňky opakují v krystalové mřížce se stejnou frekvencí. To znamená, že obě základní buňky mohou snadněji popisovat stejnou periodicitu minimálně v těchto 3 směrech. Je tak o něco pravděpodobnější, že popisují stejný krystal, pouze s jinak označenou periodou (tedy části, která se opakuje).

5.4 Měřítko kvality výsledků

K měření kvality výsledku budeme používat 2 měřítka. Prvním je pro kolik testovacích dat bylo nalezeno správné řešení. Druhým měřítkem je, na kolikáté pozici v seznamu potenciálních řešení se nachází to správné, což je obecně velmi důležité pro nejsnazší identifikaci krystalové mřížky. Dále budeme sledovat průměrný čas výpočtu. Ke zjednodušení prezentace a interpretace výsledků budeme slučovat tyto hodnoty do jedné pro všechny typy mřížek.

V tabulkách průměrnou pozici uvádíme zprůměrovanou pouze z vyřešených záznamů, tj. není žádná penalizace na tuto hodnotu za nevyřešení úlohy.

5.5 Počáteční volba parametrů

Pomocí pilotních testů byly zvoleny vhodné počáteční parametry algoritmu, které budou použité pro zjištění, zda navrhnuté změny algoritmu jsou přínosné. Počáteční nastavení je tedy toto:

- $\epsilon = 0{,}5$

- $m_{\max} = n_{\max} = 2$
- $m_C = n_C = 3$
- $LS_{hkl} = 7$
- $R_{\text{freq}} = 3$
- $Z_{\text{top}} = 200$
- $D_{\text{best}} = 4$

Parametry budou v dalších sekcích optimalizovány pro nejlepší nalezenou formu algoritmu. Cílem je najít co nejlepší výchozí nastavení pro obecné použití.

5.6 Měření kvalitativního přínosu navrhnutých změn

Tabulka 5.1: Kvalita řešení při použití navrhnutých změn

Algoritmus	Počet vyřešených	Průměrná pozice	Průměrný čas
Základní verze	81	1,11	0,313 s
+Sekce 2.4.1	90	1,19	0,556 s
+Sekce 2.4.2 C_1	91	1,24	0,556 s
−Sekce 2.4.2 C_2	90	1,23	0,565 s
−Sekce 2.4.2 C_3	90	1,29	0,560 s
+Sekce 2.4.3	93	1,17	0,565 s
−Sekce 2.4.4	92	1,33	1,190 s

V tabulce 5.1 lze vidět naměřené hodnoty. Počet vyřešených znamená, pro kolik testovacích dat obsahuje výstup mřížku, která splňuje kritérium pro správnost řešení. Průměrná pozice označuje pozici, na které se řešení nachází v průměru. Průměrný čas značí průměrnou dobu jednoho běhu výpočtu. V tabulce znaménko + pak značí, že změna byla ponechána pro další měření, znaménko − naopak znamená, že změna nebyla ponechána.

Z tabulky lze vyčíst, že přidáním hodnot $\frac{Q}{4}$ atd. podle sekce 2.4.1 došlo k výraznému zlepšení počtu vyřešených záznamů. Doba výpočtu trvala o něco déle. Vzhledem k tomu, o kolik narostl počet vyřešených úloh, je zřejmé, že tato změna se vyplatí. Ke zvýšení průměrné pozice pravděpodobně došlo kvůli tomu, že je nyní uvažováno mnohem více zón, což může vést k nalezení nesprávných mřížek, které ale odpovídají datům. Dalším možným důvodem je, že pozici zvýšily nově vyřešené záznamy.

Další návrh, změna výpočtu pro kvalitu zóny (sekce 2.4.2), také vedl ke zlepšení, alespoň pro C_1. Důvody k mírnému zvýšení průměrné pozice mohou

být stejné, jako u předchozího návrhu. Návrhy C_1 a C_2 vedly k mírnému zhoršení výsledků.

Návrh pro změnu úprav zón a mřížek ze sekce 2.4.3 vedl opět k výraznému zlepšení, tentokrát jak v počtech vyřešených, tak v průměrné pozici řešení. Časová složitost se téměř nezměnila a může být v toleranci chyby, změna tedy byla samozřejmě ponechána.

Návrh ze sekce 2.4.4 vedle ke zhoršení a nebude tedy použit ve výchozím nastavení. Nicméně je možné, že jeho uplatnění by mohlo pomoci při řešení některých méně kvalitních dat.

5.7 Finální optimalizace parametrů

Nyní budeme testovat různé hodnoty parametrů pro vybranou verzi algoritmu, tedy po uplatnění návrhů ze sekcí 2.4.1, 2.4.2 a 2.4.3.

5.7.1 Parametr ϵ

Vývoj měřítek kvality výsledků pro různé hodnoty ϵ lze vidět v tabulce 5.2. Jako nejlepší se jeví původní hodnota 0,5 a to výrazně. Pravděpodobně nejlépe odpovídá nepřesnostem při měření testovacích dat.

Tabulka 5.2: Kvalita řešení pro různé hodnoty ϵ

hodnota ϵ	Počet vyřešených	Průměrná pozice	Průměrný čas
0,1	82	1,27	0,327 s
0,3	90	1,17	0,504 s
0,5	93	1,19	0,576 s
0,7	91	1,30	0,736 s
1,0	89	1,17	1,104 s
3,0	88	2,06	5,925 s

5.7.2 Parametry $m_{\max}, n_{\max}$ a $R_{\mathbf{freq}}$

Parametry $m_{\max}, n_{\max}$ a R_{freq} jsou spjaté tím, že pokud zvýšíme $m_{\max}, n_{\max}$, získáme tím více R hodnot a tedy je vhodné měnit i R_{freq}. Vývoj pro různé trojice těchto parametrů lze vidět v tabulce 5.3. Nejlépe opět dopadla trojice původních parametrů. V ostatních případech pravděpodobně došlo k zaplnění zónami, které sice teoreticky datům odpovídají, ale nevedou ke správnému řešení.

5.7.3 Parametry m_C a n_C

V tabulce 5.4 lze vidět výsledky pro různé hodnoty m_C a n_C. Nejlépe dopadly původní hodnoty $m_C = n_C = 3$. Hodnoty $m_C = n_C = 2$ pravděpodobně

Tabulka 5.3: Kvalita řešení pro různé hodnoty $m_{\max}, n_{\max}$ a R_{freq}

$m_{\max}$	$n_{\max}$	R_{freq}	Počet vyřešených	Průměrná pozice	Průměrný čas
2	2	2	83	1,12	3,530 s
2	2	3	93	1,17	0,565 s
3	3	5	91	1,10	0,726 s
4	4	9	88	1,20	0,954 s

nedokážou dostatečně rozlišit zóny a $m_C = n_C = 4$ již pravděpodobně častěji zvyšují ohodnocení náhodou nalezeným zónám, než těm, které více odpovídají skutečnosti.

Tabulka 5.4: Kvalita řešení pro různé hodnoty m_C a n_C

m_C	n_C	Počet vyřešených	Průměrná pozice	Průměrný čas
2	2	92	1,24	0,479 s
3	3	93	1,17	0,565 s
4	4	92	1,13	0,605 s

5.7.4 Parametr LS_{hkl}

V tabulce 5.5 lze vidět výsledky pro různé hodnoty LS_{hkl}. Nejlépe dopadla původní hodnota 7. Do jisté míry je překvapivé, že se hodnota liší od m_C a n_C. Důvodem je pravděpodobně to, že zóny odpovídající skutečnosti budou odpovídat skutečným reflexím a tedy přizpůsobení i pro větší LS_{hkl} vede k přesnějším parametrům. Zatímco náhodně nalezené zóny se tak přizpůsobí pro Q hodnoty s větším m, n a výsledně dosáhnou nižší kvality.

Tabulka 5.5: Kvalita řešení pro různé hodnoty LS_{hkl}

LS_{hkl}	Počet vyřešených	Průměrná pozice	Průměrný čas
2	91	1,18	0,413 s
3	92	1,25	0,492 s
5	93	1,20	0,507 s
7	93	1,17	0,565 s

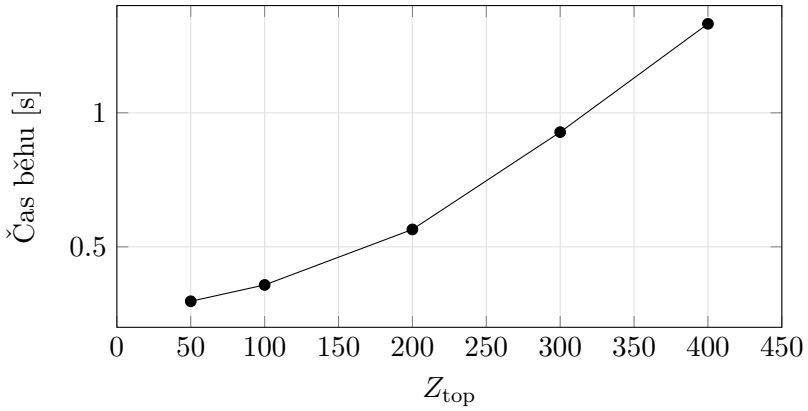
5.7.5 Parametr Z_{top}

V tabulce 5.6 lze vidět výsledky pro různé hodnoty Z_{top}. Pro hodnotu $Z_{\text{top}} = 400$ se konečně podařilo vyřešit všechny záznamy v testovacích datech. Proto i přes výrazné zvýšení doby výpočtu bude tato hodnota zachována jako výchozí.

Tabulka 5.6: Kvalita řešení pro různé hodnoty Z_{top}

Z_{top}	Počet vyřešených	Průměrná pozice	Průměrný čas
50	84	1,13	0,297 s
100	88	1,17	0,358 s
200	93	1,17	0,565 s
300	93	1,16	0,928 s
400	94	1,20	1,332 s

Zajímavý je pak vývoj času výpočtů pro různé parametry. Ten lze vidět vynesený na obrázku 5.1. Doba výpočtu roste přibližně kvadraticky, což odpovídá narůstu dle počtu testovaných dvojic zón. Zároveň však, vezmeme-li v úvahu výrazně vyšší složitost v případě, kdy dvojice má společnou Q hodnotu (proběhně navíc výpočet D hodnot – výpočet 40 hodnot a jejich zpracování navíc) oproti případu kdy není společná Q hodnota (proběhne pouze výpočet a porovnání 2×4 hodnot), lze tvrdit, že poměr dvojic se společnou Q hodnotou ku dvojicím bez společné Q hodnoty zůstává pravděpodobně přibližně stejný i pro méně kvalitní zóny.



Obrázek 5.1: Čas běhu pro různé hodnoty Z_{top}

5.7.6 Parametr D_{best}

V tabulce 5.7 lze vidět výsledky pro různé hodnoty D_{best}. Jako nejlepší se jeví opět původní hodnota, která odpovídá i hodnotě doporučené v [15].

5.8 Výchozí nastavení parametrů

Výchozí parametry algoritmu jsou tedy nastaveny na základě testování v předchozí sekci následovně:

Tabulka 5.7: Kvalita řešení pro různé hodnoty D_{best}

D_{best}	Počet vyřešených	Průměrná pozice	Průměrný čas
1	93	1,26	0,927 s
2	93	1,14	1,078 s
4	94	1,20	1,332 s
6	94	1,24	1,627 s
8	94	1,24	1,909 s

- $\epsilon = 0{,}5$
- $m_{\max} = n_{\max} = 2$
- $m_C = n_C = 3$
- $LS_{hkl} = 7$
- $R_{\text{freq}} = 3$
- $Z_{\text{top}} = 400$
- $D_{\text{best}} = 4$

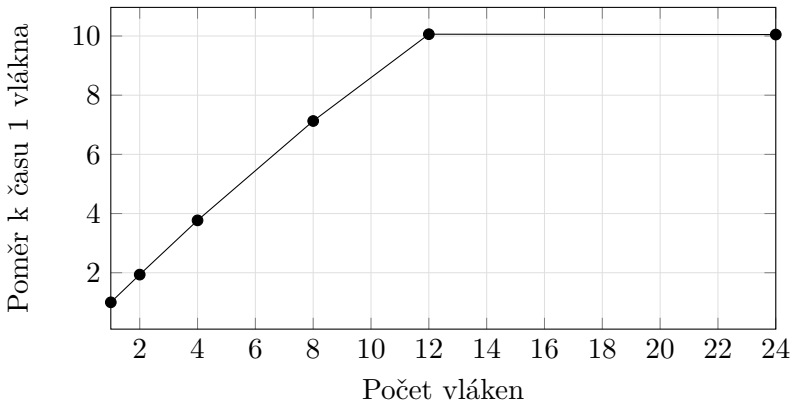
5.9 Škálovatelnost implementace

Na obrázku 5.2 lze vidět poměr $\frac{\text{doba běhu 1 vlákna}}{\text{doba běhu } n \text{ vláken}}$. Lze si všimnout, že mezi 12 a 24 vlákny nebyl v době výpočtu žádný rozdíl. Lze tedy tvrdit, že implementace nedokáže využít technologie Hyper-Threading. Jinak jsou dosažené hodnoty velmi dobré, pro 12 vláken byl výpočet proveden 10krát rychleji. Obecně lze na grafu vidět téměř lineární závislost. Konkrétní dosažené časy lze vidět v tabulce 5.8.

Vzhledem k téměř lineárnímu zrychlení lze prohlásit, že paralelizace metody proběhla úspěšně.

Tabulka 5.8: Čas výpočtu pro různé počty vláken

Počet vláken	Průměrný čas
1	10,683 s
2	5,517 s
4	2,834 s
8	1,499 s
12	1,062 s
24	1,063 s



Obrázek 5.2: Škálovatelnost implementace

5.10 Paměťová náročnost

Paměťová náročnost implementace se při zvoleném výchozím nastavení pro reálná data pohybuje okolo 200 MB.

5.11 Vybrané existující programy

K porovnání s výsledky této práce byly zvoleny 2 existující programy, které jsou v podobné kategorii časové náročnosti.

- ITO13 je originální implementace metody, dostupná z [20].
- ParaCell — metoda TREOR

K smysluplnému porovnání byly pro metodu TREOR zvoleny parametry, při kterých je čas výpočtu v řádech několika sekund.

5.11.1 Porovnání s vybranými existujícími programy

Metoda ITO13 požaduje minimálně 20 reflexí v difrakčním záznamu. Všechny kubické a hexagonální testovací data však mají méně než 20, proto je počet vyřešených mřížek pro tyto typy nulový. Nicméně pro porovnání budeme předpokládat, že by byly vyřešeny perfektně. Srovnání výsledků této práce se zmíněnými programy lze vidět v tabulce 5.9. Tato implementace dosáhla největšího počtu vyřešených záznamů a v přijatelném čase.

Dále bylo vzhledem k přizpůsobení implementace datům provedeno měření na dalších náhodně vybraných 32 triklinických záznamech z [16]. Výsledek lze vidět v tabulce 5.10. Výsledky dopadly podobně jako na testovacích datech. Této implementaci se nepodařilo vyřešit pouze 1 záznam. Zvýšený čas a pozice u všech programů je způsobena tím, že se jednalo pouze o triklinické mřížky,

Tabulka 5.9: Porovnání s vybranými programy — testovací data

Program	Počet vyřešených	Průměrná pozice	Průměrný čas
ParaCell — ITO	94	1,20	1,332 s
ParaCell — TREOR	86	1,29	10,023 s
ITO13	74	1,04	∼0,05 s

které jsou výpočetně nejnáročnější, vzhledem k tomu, že je potřeba určit všech 6 parametrů krystalové mřížky.

Tabulka 5.10: Porovnání s vybranými programy — náhodně vybraná data

Program	Počet vyřešených	Průměrná pozice	Průměrný čas
ParaCell — ITO	31	2,97	1,708 s
ParaCell — TREOR	25	3,28	26,968 s
ITO13	17	1,47	∼0,05 s

5.12 Indexing benchmark

V benchmarku dostupném z [21] si implementace vede podobně jako ITO13 s tím rozdílem, že test E3 zvládla s výchozím nastavením.

Závěr

Cílem této práce byla implementace metody ITO do programu ParaCell. Tato metoda byla implementována a kvalitativně vykazovala uspokojivé výsledky. Největší slabinou metody, jak uvádí i sám J. W. Visser, jsou nekvalitní data obsahující spoustu šumu. Proto byla implementace a její parametry přizpůsobena za pomocí kvalitních dat, na která je algoritmus cílen. Přizpůsobení bylo provedeno úspěšně a nakonec se podařilo správně oindexovat všechna testovací data.

Bylo navrhnuto několik změn pro vylepšení algoritmu a většina z nich byla aplikována, protože zlepšila kvalitu výsledků. Implementace byla paralelizována a paralelizace vykazovala téměř lineární zrychlení, což je velmi dobrý výsledek. Na závěr byla provedena optimalizace různých výchozích parametrů implementace, která vedla opět k mírnému zlepšení výsledků.

Implementace byla porovnána s 2 vybranými programy na testovacích datech. V porovnání si vedla uspokojivě. Dále bylo provedeno porovnání na náhodně vybraných 32 triklinických záznamech, kde si implementace vedla podobně. Nepodařilo se vyřešit pouze jediný záznam, což považujeme za dobrý výsledek.

Také byly provedeny experimenty na 1 indexovacím benchmarku. Tam si implementace nevedla dobře, což ale není překvapivé. Data obsahují šum a nečistoty, které jak už bylo zmíněno, působí metodě ITO problémy.

Literatura

[1] Crystallography. Direct and reciprocal lattices. https://www.xtal.iqfr.csic.es/Cristalografia/parte_04-en.html, [cit. 2022-05-03].

[2] Waseda, Y.; Matsubara, E.; Shinoda, K.: *X-Ray Diffraction Crystallography: Introduction, Examples and Solved Problems*. Springer Berlin Heidelberg, 2011, ISBN 9783642166358. Dostupné z: https://books.google.cz/books?id=vk9fnLH56DYC

[3] File:Bragg's law.svg - Wikimedia Commons. https://commons.wikimedia.org/wiki/File:Bragg%27s_law.svg, [cit. 2022-05-04].

[4] Introduction to X-ray Powder Diffraction Analysis. http://www.polycrystallography.com/XRDanalysis.html, [cit. 2022-05-04].

[5] Pecharsky, V.; Zavalij, P.: *In Fundamentals of Powder Diffraction and Structural Characterization of Materials*. 01 2003, ISBN 1-4020-7365-8, doi:10.1007/978-0-387-09579-0.

[6] Krystalické a amorfní látky. Dostupné z: http://fyzika.jreichl.com/main.article/view/622-krystalicke-a-amorfni-latky, [cit. 2021-06-27].

[7] Učebnice mineralogie pro bakalářské studium. Dostupné z: http://mineralogie.sci.muni.cz/obsah_uceb.htm, [cit. 2021-06-27].

[8] Rentgenové záření :: MEF. http://fyzika.jreichl.com/main.article/view/540-rentgenove-zareni, [cit. 2022-05-04].

[9] X-ray Powder Diffraction (XRD). https://serc.carleton.edu/msu_nanotech/methods/XRD.html, [cit. 2022-05-04].

[10] Werner, P.-E.; Eriksson, L.; Westdahl, M.: TREOR, a semi-exhaustive trial-and-error powder indexing program for all symmetries. *Journal of Applied Crystallography*, ročník 18, č. 5, Oct 1985: s. 367–370, doi: 10.1107/S0021889885010512. Dostupné z: https://doi.org/10.1107/S0021889885010512

[11] Louer, D.; Boultif, A.: Indexing with the successive dichotomy method, DICVOL04. *Zeitschrift fur Kristallographie Supplements*, ročník 2006, 06 2006, doi:10.1524/zksu.2006.suppl_23.225.

[12] De Wolff, P. M.: A simplified criterion for the reliability of a powder pattern indexing. *Journal of Applied Crystallography*, ročník 1, č. 2, 1968: s. 108–113, doi:10.1107/S002188986800508X. Dostupné z: https://onlinelibrary.wiley.com/doi/abs/10.1107/S002188986800508X

[13] Smith, G. S.; Snyder, R. L.: F_N: A criterion for rating powder diffraction patterns and evaluating the reliability of powder-pattern indexing. *Journal of Applied Crystallography*, ročník 12, č. 1, Feb 1979: s. 60–65, doi:10.1107/S002188987901178X. Dostupné z: https://doi.org/10.1107/S002188987901178X

[14] Klouda, K.: MARAST — Blog: Metoda nejmenších čtverců: řešení rovnic, které nemají řešení. https://marast.fit.cvut.cz/cs/blog_posts/19, [cit. 2022-05-04].

[15] Visser, J. W.: A fully automatic program for finding the unit cell from powder data. *Journal of Applied Crystallography*, ročník 2, č. 3, Aug 1969: s. 89–95, doi:10.1107/S0021889869006649. Dostupné z: https://doi.org/10.1107/S0021889869006649

[16] American Mineralogist Crystal Structure Database. Dostupné z: http://rruff.geo.arizona.edu/AMS/amcsd.php, [cit. 2021-06-27].

[17] ParaCell download — SourceForge.net. https://sourceforge.net/projects/paracell/, [cit. 2021-06-27].

[18] OpenMP 5.0 API C/C++ Syntax Reference Guide. 2018, [cit. 2022-04-30]. Dostupné z: https://www.openmp.org/wp-content/uploads/OpenMPRef-5.0-111802-web.pdf

[19] What Is CUDA — NVIDIA Official Blog. https://blogs.nvidia.com/blog/2012/09/10/what-is-cuda-2/, [cit. 2022-05-04].

[20] Indexing powder patterns, part 1. http://www.cristal.org/DU-SDPD/semaine-2/sdpd-2.html, [cit. 2022-05-01].

[21] Powder Diffraction Indexing Benchmarks. http://www.cristal.org/uppw/benchmarks/, [cit. 2022-05-03].

Příloha A

Seznam použitých zkratek

CPU Central processing unit

GPU Graphics processing unit

RAM Random access memory

Příloha **B**

Obsah přiloženého CD

- `src`
  - `impl` ..... zdrojové kódy implementace
  - `thesis` ..... zdrojová forma práce ve formátu LaTeX
- `text` ..... text práce
  - `DP_Cermak_Michal_2022.pdf` ..... text práce ve formátu PDF