



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra radioelektroniky

Modelování jednoosých optických krystalů

Modeling of uniaxial Optical Crystals

diplomová práce

Studijní program: Komunikace, multimédia a elektronika

Studijní obor: Multimediální technika

Vedoucí práce: doc. Mgr. Petr Páta Ph.D.

Bc. Jan Jílek

Praha 2017

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Jílek** Jméno: **Jan** Osobní číslo: **372618**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra radioelektroniky**
Studijní program: **Komunikace, multimédia a elektronika**
Studijní obor: **Multimediální technika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Modelování jednoosých optických krystalů

Název diplomové práce anglicky:

Modelling of Uniaxial Optical Crystals

Pokyny pro vypracování:

Seznamte se s principy modelování průchodu paprsku prostředím s anizotropním indexem lomu. V jazyce C/C++ vytvořte model takového prostředí, včetně energetické bilance. Navržený model by měl umožňovat definování 3D tvaru objektu s ohledem na krystalografickou orientaci a nalezení efektivních rozměrů s ohledem na využití krystalu.

Seznam doporučené literatury:

- [1] B. Stroustrup, Programovací jazyk C++, BEN, Praha, 1997.
- [2] B. Jahne, Practical Handbook on Image Processing for Scientific Applications, CRC, New York, 1997.
- [3] B. E. A. Saleh, M. C. Teich, Základy fotoniky I. - IV. Matfyzpress, MFF UK, Praha, 1994.
- [4] P. Pata, M. Klima, J. Bednar, P. Janout, C. Barta, R. Hasal, L. Maresi, and S. Grabarnik, "OFT sectorization approach to analysis of optical scattering in mercurous chloride single crystals," Opt. Express 23, 21509-21526 (2015)

Jméno a pracoviště vedoucí(ho) diplomové práce:

doc. Mgr. Petr Páta Ph.D., katedra radioelektroniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **29.09.2016** Termín odevzdání diplomové práce: **26.05.2017**

Platnost zadání diplomové práce: **20.02.2018**

Podpis vedoucí(ho) práce

Podpis vedoucí(ho) ústavu/katedry

Podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

Datum převzetí zadání

Podpis studenta

Abstrakt

Tato diplomová práce se věnuje problematice jednoosých krystalů a problematice výřezu z nich. Teoretická část shrnuje vlastnosti světla, jeho změny při průchodu krystalem samotným, a taktéž podává přehled metod pěstování krystalů pro průmyslové využití a základní vlastnosti těchto krystalů. Dále práce obsahuje základní popis nástrojů pro simulaci v jazyce C++ v grafickém rozhraní OpenGL.

V navazující praktické části tyto znalosti byly použity pro vytvoření programu, který umožňuje simulaci řezu krystalem a následný export parametru řezů do formátu pdf. Program dovoluje uživateli najít optimální řešení pozice řezu. Uživatel má možnost si toto umístění přizpůsobit nebo si výsledky uložit do externího souboru.

Klíčová slova

Boule; krystal; Sample; dvojlom; anizotropie; C++; OpenGL; desktopová aplikace; numerické řešení;

Abstract

This diploma thesis deals with the problem of uniaxial crystals and the issue of cut-out from them. The theoretical part summarizes the properties of light, its changes in the passage through the crystal itself, and provides an overview of the methods of crystal growth for industrial usage and the basic properties of these crystals. In addition, the paper contains a basic description of tools for C ++ simulation in OpenGL graphics.

In the follow-up practical part, this knowledge is used to create a program that allows the crystal cut simulation and subsequent export of the cut-off parameter into pdf format. The program allows the user to find the optimal cutting position. Users can customize this location or save the results to an external file.

Keywords

Boule; crystal; Sample; Birefringent; Anisotropy; C ++; OpenGL; Desktop application; Numerical solution;

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením doc. Mgr. Petra Páty Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Bc. Jan Jílek

PODĚKOVÁNÍ

Touto cestou bych chtěl poděkovat vedoucímu diplomové práce, doc. Mgr. Petrovi Pátovi Ph.D. za vedení, cenné poznámky, odborné připomínky a náměty pro zpracování práce. V neposlední řadě bych chtěl poděkovat mé rodině za všechno, co se nedá vyjádřit jen několika slovy.

Obsah

Obsah	6
1. ÚVOD	7
2. POPIS PROBLEMATIKY	8
2.1 Světlo.....	8
2.1.1 Elektromagnetické záření	8
2.1.2 Elektromagnetická vlna	9
2.1.3 Jevy na rozhraní	10
2.1.4 Polarizace.....	13
2.1.5 Rovnice polarizační elipsy.....	13
2.1.6 Typy polarizace.....	14
2.1.7 Anizotropní prostředí.....	15
2.1.8 Indexový elipsoid.....	16
2.1.9 Dvojlom	17
2.1.10 Energetická bilance prostředí.....	18
2.2 Krystaly	19
2.2.1 Typy krystalů	20
2.2.2 Růst krystalu	21
2.2.3 Štěpnost krystalu.....	27
2.2.4 Millerovy indexy.....	28
2.3 Nástroje pro vývoj simulací	30
2.3.1 Jazyk C++	30
2.3.2 Výrazy a přiřazení.....	30
2.3.3 Preprocesor	31
2.3.4 Funkce.....	31
2.3.5 Ukazatele	32
2.3.6 Knihovna GLM.....	32
2.3.7 Knihovna GLEW	32
2.3.8 Framework Qt.....	33

2.3.9	Knihovna OpenGL.....	34
3.	Návrh a realizace programu.....	36
3.1	Postup vývoje v jazyce C++.....	36
3.1.1	Třída MainWindow.....	37
3.1.2	Třída GLWidget.....	37
3.1.3	Třída Sample.....	37
3.2	Popis funkce programu	38
4.	Závěr	42
5.	Zdroje.....	43

1. ÚVOD

V současné době je velká poptávka po krystalech s vysokou kvalitou a dobrými optickým vlastnostmi. Optické čočky, sondy a lasery z krystalů mají širokou škálu použití v optice, elektrooptice a radiotechnice. Pro výrobu takovýchto zařízení, je potřeba krystal napřed zpracovat, např. vyříznout a vybrousit optickou čočku. Aby nedocházelo ke zbytečnému plýtvání tímto drahým materiálem je nezbytné, napřed tento výřez v krystalu nasimulovat. Jazyk C++ je vhodný nástroj pro tvorbu takového simulačního programu díky své procedurální rychlosti, snadné implementaci grafického rozhraní pomocí OpenGL a také jazyk C++ není omezen jen na jednu platformu.

Cílem této diplomová práce je seznámit se s principy modelování průchodu paprsku prostředím s anizotropním indexem lomu. Dalším cílem bylo navrhnout a naprogramovat takové prostředí, které by umožňovalo definovat 3D tvar s ohledem na krystalografickou orientaci krystalu.

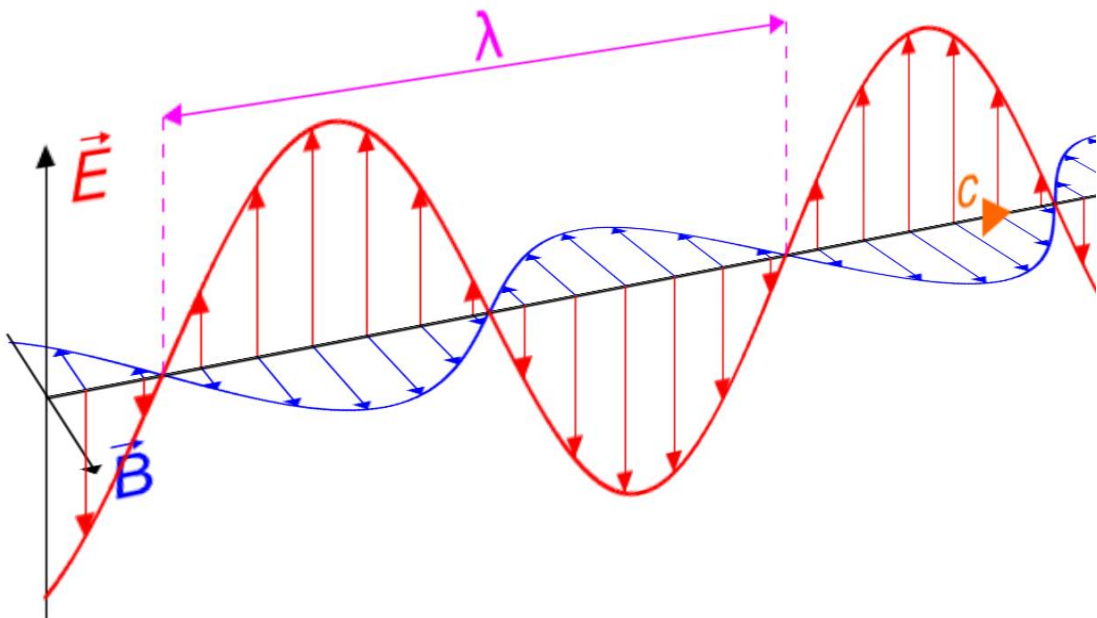
2. POPIS PROBLEMATIKY

2.1 Světlo

Světlo je viditelná část elektromagnetického záření, na kterou je citlivý lidský zrakový orgán – oko. Viditelné světlo je obvykle definováno jako vlnové délky v rozmezí 400 – 700 nm a leží mezi infračerveným (s delšími vlnovými délkami) a ultrafialovým zářením (s kratšími vlnovými délkami) [1]. Světlo může být popsáno buď jako elektromagnetická vlna, která má konstantní rychlost šíření ve vakuu, nebo jako proud fotonů – částic, které vykazují určitou energii, hybnost, vnitřní moment hybnosti a nulovou hmotnost. Primárními vlastnostmi viditelného světla jsou rychlost, intenzita, směr šíření, kmitočtové spektrum nebo spektrum vlnových délek a polarizace.

2.1.1 Elektromagnetické záření

Elektromagnetické záření je kombinace dvou vektorových polí, magnetického a elektrického. Tato pole jsou navzájem kolmá a jejich kmity probíhají napříč ke směru, kterým se vlnění šíří. Na Obr. 1 se elektromagnetická vlna šíří v kladném směru podél osy z, vektor intenzity elektrického pole se promítá do osy y a vektor magnetické indukce do osy x.



Obr. 1 Elektromagnetické záření[2]

Elektrické pole, které se pohybuje nebo se mění, vyvolá magnetické pole a měnící se nebo pohybující se magnetické pole vyvolává pole elektrické (vychází z Maxwellových rovnic). Za hlavní charakteristiky elektromagnetického záření jsou považovány frekvence, vlnová délka a polarizace. Na Obr. 1 je zobrazena vlnová délka λ , která označuje nejbližší vzdálenost mezi dvěma body vlnění, které kmitají se stejnou fází. Vlnovou délku λ lze vypočítat pomocí vztahu

$$\lambda = v \cdot T = \frac{v}{f} [m] \quad (1)$$

- T – perioda
- F – frekvence vlnění
- V – rychlost šíření vlny

A také rychlost světla c

$$c = \frac{1}{\sqrt{\varepsilon_0 \mu_0}} \approx 3 \cdot 10^8 [m/s] \quad (2)$$

- ε_0 – permitivita vakua ($8,85 \cdot 10^{-12} [Fm^{-1}]$)
- μ_0 – permeabilita vakua ($1,26 \cdot 10^{-6} [Hm^{-1}]$)

2.1.2 Elektromagnetická vlna

Vlnový charakter elektromagnetického záření lze popsat pojmem elektromagnetická vlna. Podstatu elektromagnetického vlnění popsal ve 2. polovině 19. století skotský fyzik James Clarc Maxwell. Elektromagnetickou vlnu v látkovém prostředí popisují tzv. Maxwellovy rovnice v látkovém prostředí bez volných nábojů a proudů

$$rot \vec{H} = \frac{\partial \vec{D}}{\partial t} \quad (3)$$

$$rot \vec{E} = -\frac{\partial \vec{B}}{\partial t} \quad (4)$$

$$div \vec{D} = 0 \quad (5)$$

$$div \vec{B} = 0 \quad (6)$$

- \vec{D} – vektor elektrické indukce
- \vec{E} – vektor intenzity elektrického pole
- \vec{B} – vektor magnetické indukce
- \vec{H} – vektor intenzity magnetického pole

Vztah mezi elektrickou indukcí a elektrickým polem závisí na charakteru dielektrického prostředí. Podobně vztah mezi magnetickou indukcí a magnetickým polem závisí na magnetických vlastnostech prostředí. Tyto vztahy jsou definovány rovnicemi

$$\vec{D} = \varepsilon_0 \varepsilon_x \vec{E} = \varepsilon_0 \vec{E} + \vec{P} \quad (7)$$

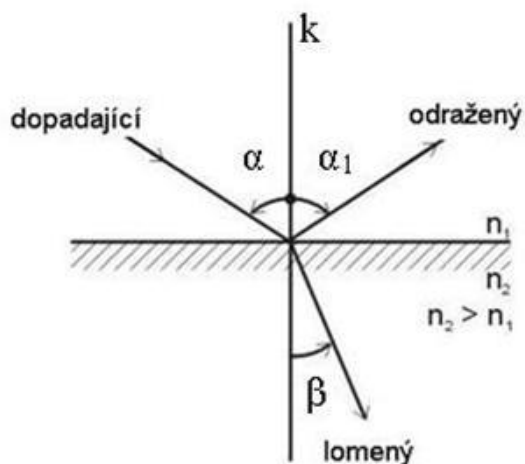
$$\vec{B} = \mu_0 \mu_x \vec{H} = \mu_0 \vec{H} + \vec{M} \quad (8)$$

- \vec{D} – vektor polarizace
- \vec{M} – vektor magnetizace

Tyto rovnice se nazývají materiálové vztahy [3].

2.1.3 Jevy na rozhraní

Na rozhraní dvou homogenních a izotropních prostředí (ve všech směrech stejné vlastnosti) se světlo částečně odráží a částečně láme do druhého prostředí (viz Obr. 2 *Jevy na rozhraní*), při tomto jevu se světelné paprsky řídí zákonem odrazu a lomu (tzv. Snellův zákon).



Obr. 2 Jevy na rozhraní[4]

2.1.3.1 Zákon odrazu

Na rovinném rozhraní dvou homogenních, izotropních a rozdílných prostředí se paprsky řídí zákonem odrazu. Ten říká, že dopadající paprsek se odráží pod stejným úhlem jako dopadl vzatým ke kolmici k rovině dopadu.

$$\alpha = \alpha_1 \quad (9)$$

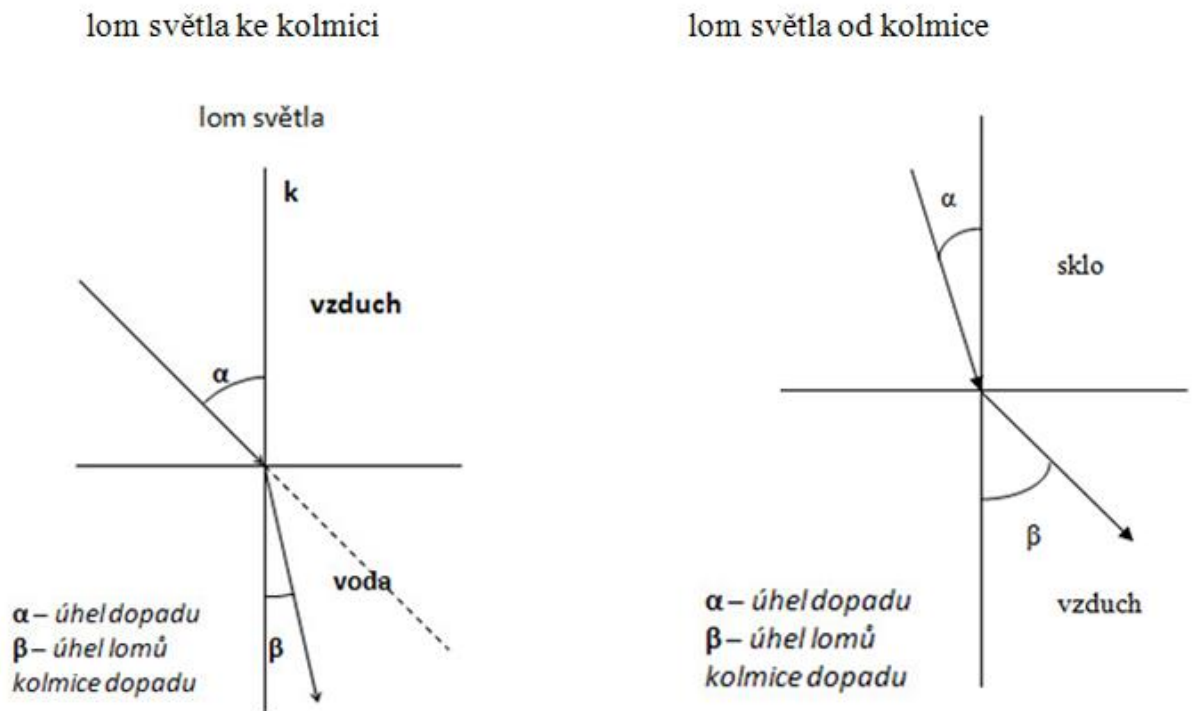
2.1.3.2 Zákon lomu

Lom vlnění neboli refrakce je změna směru šíření paprsku (vlny), která se vyskytuje na rozhraní dvou prostředí, kterými tato vlna prochází, to platí pro veškeré druhy elektromagnetického záření včetně světla. Ten jev se vysvětluje zákonem zachování energie a hybnosti. Při změně prostředí se mění rychlost vlny, a jeho frekvence zůstává stejná [5], tj. lom světla je důsledek různých rychlostí šíření světla v různých prostředí. Nejjednodušším a zřejmým příkladem deformace paprsku je lom světla sklem nebo ve vodě. Lom světla se řídí Snellovým zákonem.

Snellův zákon je jedním ze základních zákonů geometrické optiky, byl objeven v 17. století nizozemským matematikem a astronomem Willebrordem Snelliusem. Zákon popisuje, jakým způsobem se chová elektromagnetické záření na rovinném rozhraní dvou prostředí s odlišnými vlastnostmi.

Obecně platí, že na rovinném rozhraní dvou homogenních, izotropních a rozdílných prostředí s indexy lomu n_1 a n_2 se světelné paprsky řídí podle rovnice

$$n_1 \sin \alpha = n_2 \sin \beta \quad (10)$$



Obr. 3 Zákon lomu [4]

Ze Snellova zákona plynou dvě možnosti

- Při šíření záření z prostředí opticky řidšího do opticky hustšího prostředí ($n_1 < n_2$) se paprsky lámou směrem ke kolmici (tzv. lom ke kolmici)
- Při šíření záření z prostředí opticky hustšího do opticky řidšího prostředí ($n_2 < n_1$) se paprsky lámou směrem od kolmice (tzv. lom od kolmice) (viz Obr. 3).

2.1.3.3 Brewsterův úhel

$$\theta_b = \arctg \frac{n_2}{n_1} \quad (11)$$

Jedná se o úhel, při které se odráží pouze polarizovaná složka světla, zbytek se láme do druhého prostředí.

2.1.4 Polarizace

Směr světla neboli vektor, kterým světlo kmitá, je jednou z jeho nejdůležitějších vlastností. Světlo je nepolarizované, pokud je jeho vektor náhodný. S výjimkou laseru a maseru poskytují všechny primární světelné zdroje světlo nepolarizované. Pokud mluvíme o světle polarizovaném, uvažujeme, že jeho vlnění se šíří pouze v určitém směru a vektor kmitá pouze v jedné rovině [6].

Polarizace světla je určena směrem vektoru intenzity elektrického pole $\vec{E}(r, t)$ a jeho závislostí na čase. Pro monochromatické světlo se tři složky vektoru $\vec{E}(r, t)$ sinusově mění s časem. Jejich amplitudy a fáze jsou přitom obecně různé, takže se v každém místě \mathbf{r} pohybuje koncový bod vektoru $\vec{E}(r, t)$ v rovině, v níž opisuje elipsu. To je znázorněno na Obr. 4. Rovina, orientace a tvar elipsy závisí obecně na poloze. Necháme-li pouze kmity určitého směru, bude mít vlnění v různých směrech kolmých ke směru šíření různé vlastnosti, a tak půjde o světlo polarizované.

Orientace a excentricita elipsy určuje stav polarizace optické vlny, zatímco její rozměry jsou určeny intenzitou světla. Je-li elipsa degenerovaná na přímku, nazýváme vlnu lineárně polarizovanou, jestliže se stane kružnicí, nazýváme ji kruhově polarizovanou. Polarizace hraje důležitou roli v interakci světla s látkou, což dokazují následující příklady:

- Množství světla odraženého na rozhraní dvou látek závisí na polarizaci vlny
- Množství světla absorbovaného určitými materiály závisí na polarizaci
- Rozptyl světla v látce je obecně citlivý na polarizaci
- Index lomu anizotropních materiálů závisí na polarizaci. Různě polarizované vlny se tedy šíří různými rychlostmi a vykazují různá fázová posunutí, takže se při šíření vlny polarizační elipsa mění (např. lineární polarizované světlo se může změnit na světlo polarizované kruhově). Této vlastnosti se využívá při konstrukci mnohých optických zařízení [3].

2.1.5 Rovnice polarizační elipsy

Uvažujme monochromatickou rovinnou vlnu frekvence \mathbf{f} , která se šíří ve směru z s rychlostí \mathbf{c} . Vektor elektrického pole leží v rovině x - y a je obecně popsán vztahem, kde komplexní obálka je vektor s komplexními složkami. K popsání polarizace této vlny najdeme koncový bod vektoru $\vec{E}(z, t)$ v každém místě na ose z jako funkci času.

Rovnice polarizační elipsy v rovině kolmé na směr šíření

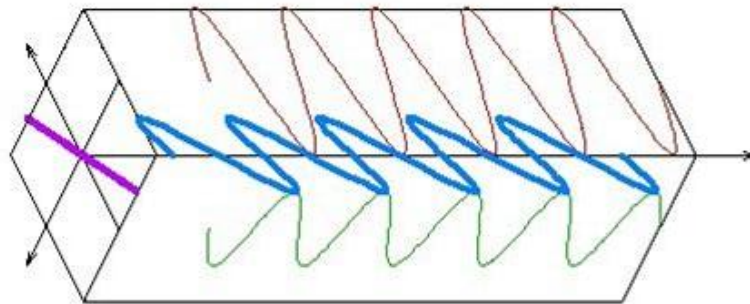
$$\frac{E_x^2}{e_x^2} + \frac{E_y^2}{e_y^2} - \frac{2E_xE_y}{e_x e_y} \cos(\varphi_y - \varphi_x) = \sin^2(\varphi_y - \varphi_x) \quad (12)$$

- E_x a E_y jsou složky vektoru elektrické intenzity ve směru příslušné osy
- e_x a e_y jsou amplitudy složek vektoru elektrické intenzity
- φ_x a φ_y jsou počáteční fáze

2.1.6 Typy polarizace

Lineární polarizace

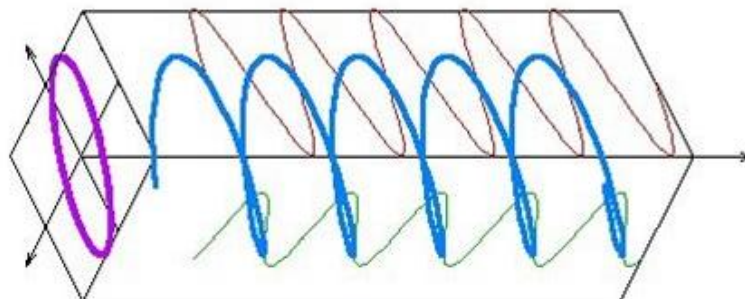
Vlna, jejichž kmitání probíhá v určité rovině, všechny výchylky leží ve všech místech vlny ve stejné rovině (viz Obr. 4) [7].



Obr. 4 Schéma lineární polarizace[8]

Eliptická polarizace

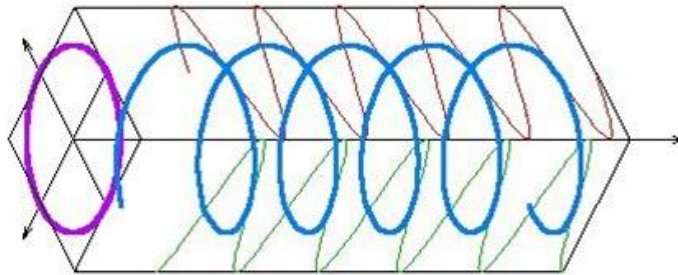
Výchylky opisují elipsu v rovině kolmé k šíření vlny (viz Obr. 5) [7].



Obr. 5 Schéma eliptické polarizace [8]

Kruhová polarizace

Speciální případ elipticky polarizované vlny. Výchylinky opisují kružnici (viz Obr. 6).



Obr. 6 Schéma kruhové polarizace [8]

Existují dvě možnosti

- Je-li $\sin\varphi < 0$ jde o pravotočivou kruhovou polarizaci
- Je-li $\sin\varphi > 0$ jedná se levotočivou kruhovou polarizaci [7]

2.1.7 Anizotropní prostředí

Prostředí, v němž se z daného bodu šíří světlo všemi směry stejnou rychlostí, nazýváme izotropní. Jeho optické vlastnosti jsou zadány jediným indexem lomu. Speciálním příkladem izotropního prostředí je vakuum.

Pokud rychlost šíření světla závisí na směru, hovoříme o prostředí anizotropním. Optické vlastnosti anizotropních prostředí popisujeme pomocí několika indexů lomu (dvou nebo tří) [9]. Příkladem opticky anizotropních prostředí jsou některé krystaly, např. krystal křemene.

Dielektrické prostředí nazýváme anizotropní, závisí-li jeho makroskopické optické vlastnosti na směru. Makroskopické vlastnosti látky jsou ovšem určeny mikroskopickými vlastnostmi: tvarem a orientací jednotlivých molekul a prostorovým uspořádáním jejich středů. Dále je uveden popis typů prostorového a směrového uspořádání, které se vyskytují v různých druzích optických materiálů.

- Jestliže jsou molekuly umístěny v prostoru zcela náhodně a jsou samy izotropní nebo jsou orientovány zcela náhodně, je prostředí izotropní. Izotropní jsou plyny, kapaliny a amorfní pevné látky

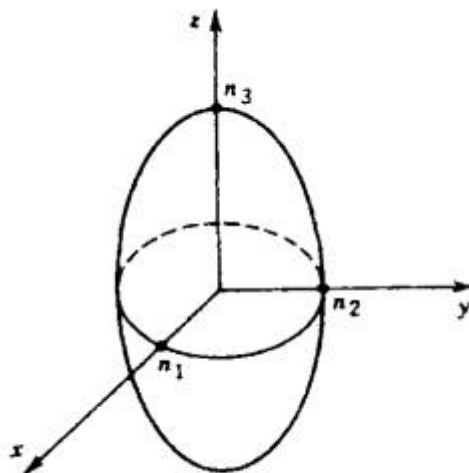
- Jestliže jsou molekuly anizotropní a nejsou zcela náhodně orientovány, je látka anizotropní, i když jsou prostorově rozmístěny zcela náhodně. To je případ kapalných krystalů, které mají směrové uspořádání, ale zcela postrádají uspořádání polohové
- Jestliže jsou molekuly uspořádány v prostoru do pravidelného periodického obrazce a jsou orientovány stejným směrem, jak je tomu v krystalech, je prostředí obecně anizotropní
- Polykrystalické materiály mají strukturu vytvořenou z těsně se dotýkajících krystalických zrn, která jsou vůči sobě náhodně orientovaná. Zrna jsou sama o sobě obecně anizotropní, ale jejich zprůměrované makroskopické chování je izotropní [3]

2.1.8 Indexový elipsoid

Indexový elipsoid reprezentuje tenzor pomocí kvadratické plochy. V systému souřadnic hlavních os, je indexový elipsoid popsán rovnicí

$$\frac{x^2}{n_1^2} + \frac{y^2}{n_2^2} + \frac{z^2}{n_3^2} = 1 \quad (13)$$

Optické vlastnosti krystalu, tedy směry v hlavních osách a index lomu podél nich, jsou zcela popsány indexovým elipsoidem. Indexový elipsoid jednoosého krystalu je rotační elipsoid a elipsoid opticky izotropní látky je kulová plocha (viz obr. 7) [10].



Obr. 7 Indexový elipsoid popsáný pomocí souřadnic hlavní osy a příslušnými indexy lomu [10]

2.1.9 Dvojlom

Klasické sklo je homogenní a izotropní, to znamená že má stejný index lomu a nezáleží jakým směrem světlo skrze ně cestuje. Jednoosé materiály, jako je krystal islandského vápence (kalcit, CaCO_3), mají osy definované s osovou symetrií. Tyto materiály lámou paprsky světla rozdílně, podle toho, jaká je zrovna polarizace paprsku a jaký je úhel paprsku vzhledem k osám krystalu.

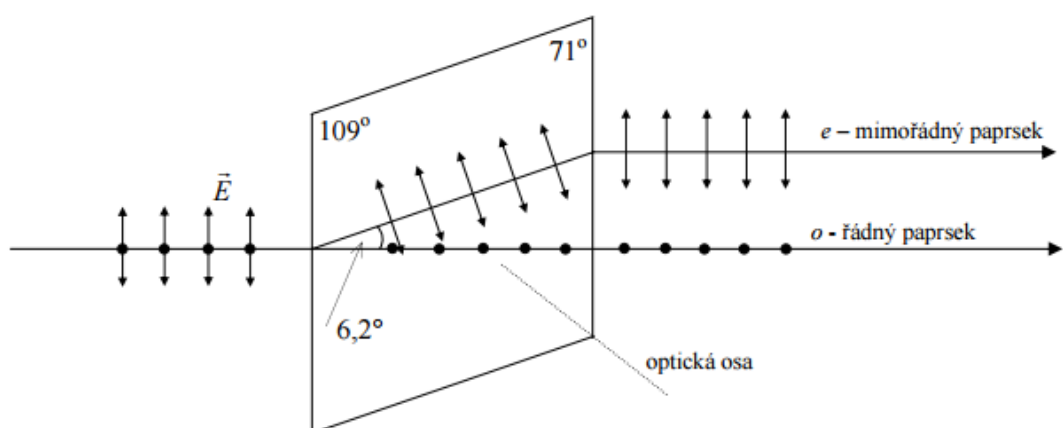
Při průchodu vzniknou dva paprsky, které se šíří různými směry – na výstupu jsou oba dva vzájemně posunuté a rovnoběžné s dopadajícím paprskem. Proto vznikají dva možné uhly lomu pro každý paprsek, reprezentující dva ortogonální polarizační stavy. Dvojlomné materiály vždy ohýbají paprsky podle Snellova zákona, ale efektivní index lomu v prostředí závisí na vstupní polarizaci paprsku a na úhlu, který svírá s krystalovou osou. Běžně je paprsek pak odražen dle

$$n \sin\theta = n_o \sin\theta' \quad (14)$$

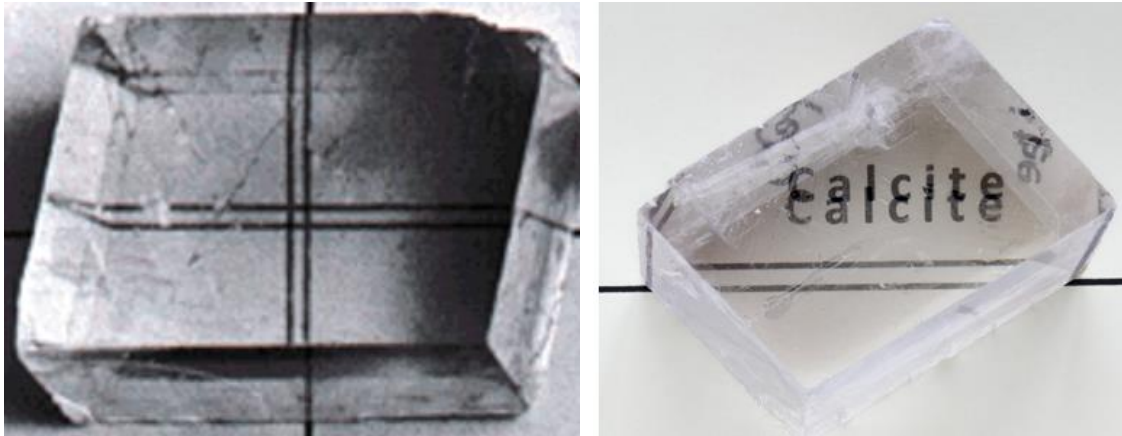
kde n_o je řádný (angl. ordinary) index lomu. Ostatní paprsky se pak lámou dle

$$n \sin\theta = n_e \sin\theta' \quad (15)$$

což je tedy také Snellův zákon, ale mimořádný (angl. extraordinary) paprsek a jeho index lomu n_e je funkce úhlu α_w , tedy úhlu mezi osou krystalu vektorem \vec{a} a lomným vlnovým vektorem \vec{k} [11].



Obr. 8 Paprsek světla se dvěma ortogonálními polarizacemi procházející rovinou kalcitu) [12]



Obr. 9 Dvojlom na kalcitu [13]

2.1.10 Energetická bilance prostředí

Amplitudové odrazivosti a propustnosti \mathbf{r} a \mathbf{t} jsou poměry komplexních amplitud. Výkonová odrazivost \mathbf{R} a propustnost \mathbf{T} jsou definovány jako poměry toku výkonu (ve směru kolmém k rozhraní) odražené a procházející vlny k toku výkonu vlny dopadající. Protože se odražená a dopadající vlna šíří ve stejném prostředí a svírá stejný úhel s normálou k povrchu platí

$$\mathbf{R} = |\mathbf{r}|^2 \quad (16)$$

Zákon zachování energie vyžaduje, aby byl splněn vztah

$$\mathbf{T} = 1 - \mathbf{R} \quad (17)$$

Je nutno však vzít v úvahu, že $\mathbf{T} = [n_2 \cos\theta_2 / n_1 \cos\theta_1] |\mathbf{t}|^2$ není obecně rovno $|\mathbf{t}|^2$, protože se výkon šíří pod různými úhly.

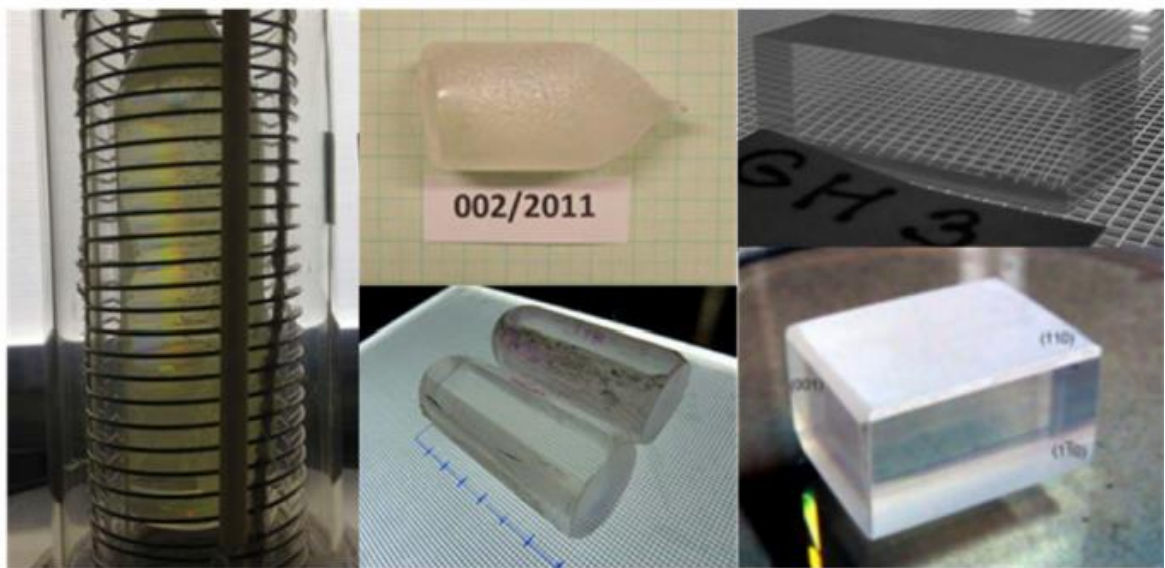
2.2 Krystaly

Každá krystalická látka může vytvářet více či méně pravidelné těleso krystalu, jehož tvar odpovídá symetrii vnitřního uspořádání stavebních částic [14].

Krystalické látky můžeme rozdělit do dvou skupin: polykrystaly a monokrystaly. Polykrystaly jsou krystalické látky, které se skládají z velkého množství drobných malých (od 10 μm do několika milimetrů) zrn krystalů, uspořádaných pravidelně – jsou to malé monokrystaly, ale vzájemná krystalografická orientace jednotlivých zrn je nahodilá [15]. Různá orientace zrn v polykrystalických materiálech způsobuje makroskopickou izotropii.

Monokrystal je opakem polykrystalických materiálů. Monokrystal má v celém svém objemu jediný stále se opakující strukturní motiv [7]. Na rozdíl od polykrystalických materiálů jsou veškeré fyzikální vlastnosti monokrystalů anizotropní.

Jako příklad je na Obr. 10 zobrazen monokrystalický calomel a jeho výbrus.



Obr. 10 Boule a sample calomel

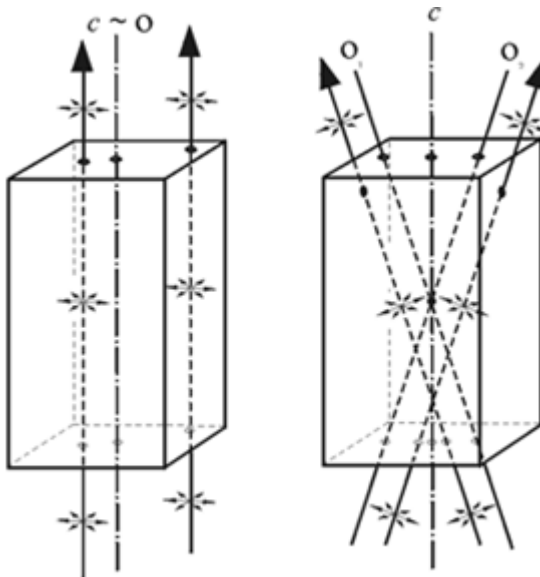
2.2.1 Typy krystalů

Podle lomu a dvojlomu světla látky se dělí na opticky izotropní a anizotropní. Anizotropní krystaly dále můžeme rozdělit na jednoosé a dvouosé.

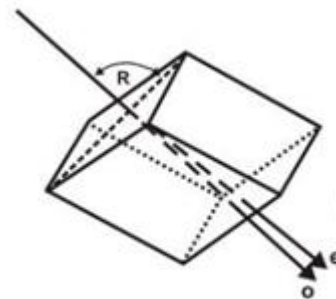


Obr. 11 Rozdělení druhů krystalu podle optických vlastností

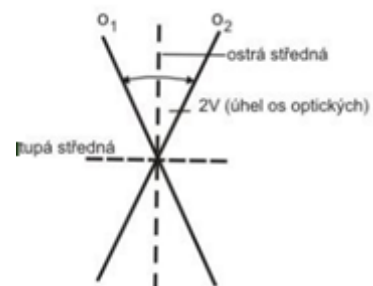
V jednoosých minerálech existuje jeden směr (optická osa), ve kterém nedochází ke dvojlomu, ale pouze k lomu, tento je souhlasný se směrem krystalografické osy C (viz Obr. 12a). Ve směrech odchýlených od optické osy dochází ke dvojlomu (viz Obr. 12b).



Obr. 12a Optické jednoosé krystaly mají jeden směr optické izotropie. Optické dvojosé krystaly mají dva směry optické izotropie



Obr. 12b Dvojlom v jednoosém krystalu



Obr. 12c Dva směry izotropií ve dvouosém krystalu

Jeden z paprsků, který vzniká dvojlomem, se chová jako paprsek v izotropním prostředí: při kolmém dopadu nepodléhá lomu a ve všech směrech se krystalem šíří stejnou rychlostí to je **paprsek řádný** (značí se o). Řádný paprsek má ve všech směrech stejný index lomu (značí se n_o). Druhý paprsek se chová odlišně i při kolmém dopadu se láme pod určitým úhlem a v různých směrech se krystalem šíří různou rychlostí. Proto se index lomu mimořádného paprsku (n_e nebo e) mění se směrem šíření světla, to je **mimořádný paprsek** (značí se ϵ) [16].

Krystaly se nazývají kladné jednoosé, jestliže $n_e > n_o$, a záporné jednoosé, jestliže obráceně.

V opticky dvojosých krystalech existují dva směry (optické osy O_1 a O_2), ve kterých se krystal chová jako opticky izotropní (nedochází k dvojlomu) [16]. Tyto osy svírají u různých minerálů různě velký úhel optických os $2V$ (viz Obr. 12c). Ve všech ostatních směrech dvojlom nastává, přitom oba vzniklé paprsky se chovají jako paprsky mimořádné (šíří se v různých směrech různou rychlostí).

2.2.2 Růst krystalu

Pro některé unikátní vlastnosti dnes monokrystaly našly své využití v radioelektronice, v kvantové elektronice, mikrovlnných technikách, akustice, apod. [7]. Zpočátku byly využívány monokrystaly přírodního původu, avšak kvůli stálému využívání monokrystalů a také nedokonalostem mřížek přírodních monokrystalů byla započata výroba umělých [17].

První pokusy s přípravou monokrystalů se začaly rozvíjet v 19. století ve Francii. První oficiálně zveřejněná metoda pro přípravu umělých rubínů byla publikována Augustem Verneilem v roce 1902. Tato metoda byla až do roku 1945 nejužívanější metodou přípravy krystalů, avšak pomocí ní byly připraveny pouze dva typy drahokamů – rubín a safír. Mezi tím v roce 1925 byla zveřejněna Bridgmanova metoda a v roce 1936 její modifikace - metoda podle Stockbargera [18].

Dnes obvykle syntetické monokrystaly se vyrábějí pěstováním z taveniny – Verneuilova a Czochralského metoda růstu krystalů. Dále se používá pěstování krystalů z roztoků anebo pěstování pomocí zónové tavby [7]. Způsob přípravy monokrystalu závisí na vlastnostech připravovaného materiálu [17]. Například pro růst monokrystalu

křemíku se používá Czochralského metoda, pro růst monokrystalů oxidů (např. beryllia oxidu hlinitého (BeAl_2O_4)) metoda plovoucí zóny [19].

V současné době existují desítky různých metod růstu krystalů (metoda horkého drátu, Schmidt-Viechnického metoda, Stockbargerova metoda, metod zonální tavby, metoda plavající zóny atd.). V této práci jsou popsány obecně uznávané technologie růstu jádra krystalů, od kterých se vyvinuly všechny ostatní procesy.

2.2.2.1 Verneuilova metoda

V roce 1902 v Paříži Auguste Verneuil začal průmyslovou výrobu krystalů, kdy použil proces růstu fúzí plamenem, pro získání jednotlivých krystalů rubínu a safíru při teplotách tání nad $2000\text{ }^\circ\text{C}$.

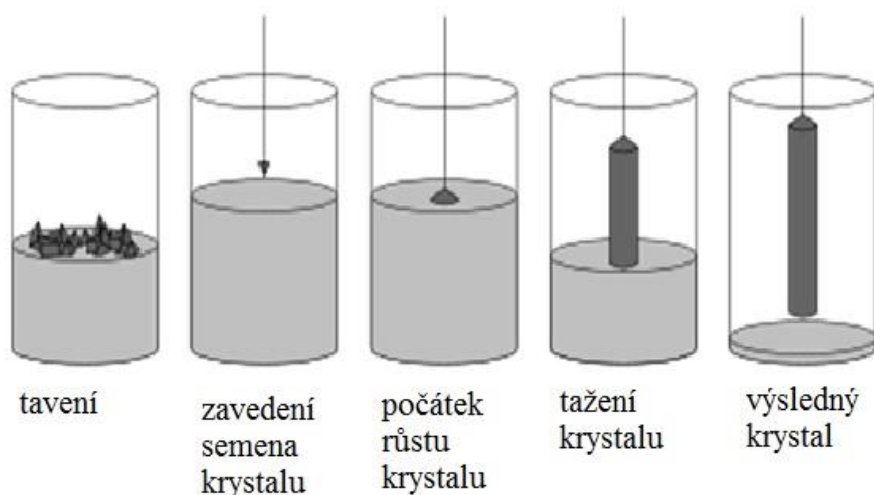
Fyzika zapojená do Verneuilova procesu se prakticky nezměnila od původního návrhu, omezuje ale velikost a tvar výrobku, které lze vyprodukovat [20]. Princip metody spočívá v tavení práškového materiálu v plamenu a ukládání kapek taveniny na povrch krystalu umístěného v temperované komoře [18]. Výsledné krystaly mají typicky zakřivené pruhy způsobené růstem, které omezují jejich použití v optických aplikacích. I dnes ale zůstává významná poptávka po safíru pěstovaném pomocí metody Verneuilovi, protože je to stále nejlevnější způsob, jak si zajistit safírový materiál vhodný pro mnoho použití [20].

2.2.2.2 Czochralského metoda

Metoda růstu krystalů Czochralského byla objevena v roce 1916 polským vědcem Janem Czochralským. Při práci v laboratoři Czochralského omyl psací pero do kelímku s roztaveným cínem a prudce ho vytáhl. Ze špičky pera visel tenký kovový závit. Czochralský zjistil, že vykryštalizovaný drát byl ve skutečnosti monokrystal [21].

Od roku 1916 Czochralského metoda byla značně vylepšena, ale v podstatě se nezměnila. Semeno krystalu je vloženo do kelímku uvnitř vlastní růstové vakuové komory. Komora je pak naplněna inertním plynem, proto, aby se zabránilo vniknutí atmosférických plynů do taveniny během růstu krystalu. Materiál se roztaví a do taveniny se ponoří tenké semeno safíru s přesnou orientací. Semeno krystalu se odebírá řízenou rychlostí, a přitom se krystal a kelímek otáčejí v opačných směrech. Proces se opakuje a krystalové vrstvy se přidávají s každým otočením, dokud není dosažena cílová velikost a tvar. Postup růstu krystalu touto metodou je znázorněn na Obr. 13.

Proces růstu může trvat až osm týdnů a vyžaduje pečlivou, nepřetržitou pomoc a monitorování. Výsledný safírový materiál má dobré optické vlastnosti a má široké uplatnění v laserové technice, infračervených a ultrafialových oknech, transparentních elektronických površích, oknech s vysokou teplotní odolností a dalších optických aplikacích [20].



Obr. 13 Postup růstu krystalu metodou Czochralski

2.2.2.3 Nacken-Kyropoulosova metoda

V roce 1926 byla vyvinuta Spyrem Kyropulosem metoda pro přímou krystalizaci taveniny. Kyropoulos představil svou techniku jako způsob výroby velkých monokrystalů, které jsou bez trhlin a poškození.

Růst krystalu probíhá na stacionárním zárodku, který je upevněn na otáčející se zárodkové - chladící tyči [18]. Čistý prášek oxidu hlinitého (Al_2O_3) je umístěn do kelímku, který se zahřeje na teplotu tání. Krystal je tvořen hluboko pod povrchem roztaveného oxidu hlinitého a ztuhne přímo do tvaru kelímku. Proces se řídí tepelným gradientem, tak aby pouze vrstva krystalů na rozhraní pevná látka-kapalina zůstala roztavena a semeno krystalu se pomalu vyťahovala z krystalových vrstev a tím velikost boule krystalu rostla.

Tato technika růstu je ideální pro materiály s nízkou tepelnou vodivostí a vysokým stupněm tepelné roztažnosti, jiná kombinace může způsobit, že křišťálový materiál bude vystaven nejružnějším nedostatkům, pokud nebude pěstován a ochlazován v prostředí s nízkým roztažným napětím.

Díky tomuto přesně řízenému tepelnému gradientu vykazují krystaly vypěstované Kyropoulosovou metodou velmi vysokou optickou kvalitu. Výsledné boule mohou být řezány do libovolné krystalografické orientace nebo roviny. Takové krystaly jsou ideální pro mnoho optických aplikací včetně elektroniky (substráty, IR (angl. Infra - red) detektory, optické vláknové čočky) a optiky (okna, čočky, sondy, lasery) [20].

2.2.2.4 Metoda výměny tepla

Metoda výměny tepla (angl. Heat Exchanger Method / HEM) používaná pro pěstování velkých safírových boulí, se často nazývá modifikovanou metodou Nacken-Kyropoulosovou.

V moderním provedení HEM je na dno kelímku umístěno semínko krystalu, kelímek je poté naplněn čistým prachem z oxidu hlinitého (Al_2O_3). Z pece je vyhnán vzduch a pec je zahřívána tak, aby roztavila všechny zbytky, přičemž se semeno udržuje těsně pod bodem tání. Toho je docíleno tím, že se přes výměník tepla pod kelímkem prohání helium. Teplo a vakuum pomáhají vyčistit oxid hlinitý odpařením nečistot. Po částečném roztavení semena se zvýší průtok helia, aby se semeno opět ochladilo a spustila se krystalizace oxidu hlinitého na osivu. Pec se udržuje při konstantní teplotě během růstu krystalu, který roste ze semen ve třech směrech. Mírné změny tepelného toku ovlivní tvar rostoucího krystalu[20]. Na Obr. 14 je zobrazen safír vypěstovaný metodou výměny tepla.



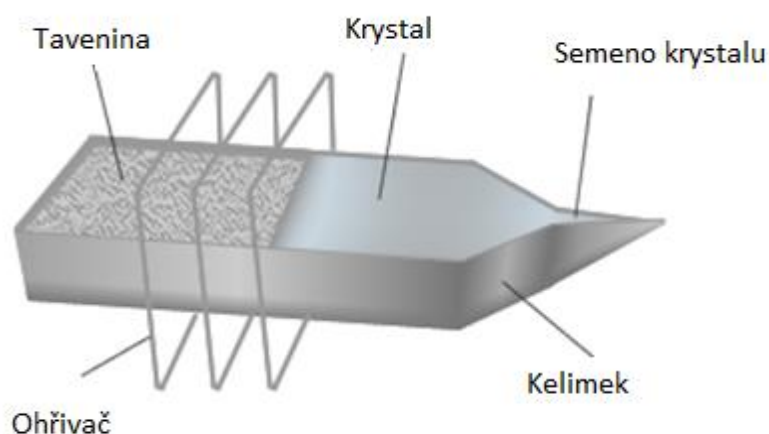
Obr. 14 Safír vzniklý metodou HEM [22]

Když je krystalizace dokončena, teplota pece a tok plynu se sníží a krystalová boule pomalu vychládá. Díky dlouhému a pomalému procesu chlazení dochází k výjimečné kvalitě krystalů. Pokusy vyrábět hodně velké boule pomocí této metody se ukázaly jako neúspěšné díky častému praskání během procesu chlazení [20].

2.2.2.5 Bagdasarova metoda (Vodorovně řízená krystalizace)

Bagdasarová metoda neboli horizontální metoda růstu krystalu (angl. Bagdasarov Horizontally Directed Crystallization Method), také známá jako horizontální Bridgmanova metoda, byla vyvinuta v roce 1964. [20]

Základem metody je umístění prášku oxidu hlinitého (Al_2O_3) a zárodečného krystalu do kelímku ve tvaru lodi a následné tažení pod wolframovým topičem. Ampule s ohřivačem je umístěná ve vakuové molybdenové peci (viz Obr. 15) [23]. Krystaly se pěstují rychlostí 8-10 mm za hodinu.



Obr. 15 Kalíšek pro horizontální metodu [24]

Tato metoda umožňuje získat velké pláty s téměř dokonalými hranami a libovolnou krystalografickou orientací. Typicky jsou krystaly pěstované tímto způsobem tvarovány jako tlusté obdélníky např. 220 mm x 250 mm x 25 mm [20].

Jednou z výhod této metody je její relativní technologická jednoduchost. Kromě toho umožňuje technika HDC rekrystalizovat surovinu před růstem a odstranit tak většinu nečistot.

Metoda HDC se používá pro pěstování safíru a rubínových monokrystalů. Skutečnost, že povrch taveniny je velmi široký ve srovnání s jeho tloušťkou, umožňuje snadné přidání jakékoliv požadované příměsi během procesu krystalizace.

Safíry vypěstované Bagdasarovou metodou mají velmi vysokou optickou kvalitu a jsou vhodné pro velmi náročné optické, mechanické a LED aplikace.

2.2.2.6 Edge-Defined Film-Fed Growth

V roce 1965 byl technik Harold Labelle zařazen do společnosti Tyco Industries ve Walthamu MA, aby vyvinul proces růstu safírových vláken jako výztuže pro jiné sloučeniny. Díky zkušenostem s experimenty s růstem krystalů pro polovodiče se mu podařilo dobře ovládnout rozhraní pevných látek a kapalin, což je klíčová koncepce růstu krystalů. Labelle zjistil, že po ponoření studeného wolframu do roztaveného oxidu hlinitého vznikají dendrity malého krystalického oxidu hlinitého na povrchu wolframu. [20]

Výsledný proces byl v podstatě poddruh metody Czochralski s výhodou kontroly tvaru krystalu. Nakonec byly přidány do procesu matrice s otvory o velikosti a tvaru požadovaného krystalu.

Dnes je technologie EFG krystalového růstu stejně jednoduchá jako ji vyvinul Labelle. Hliník se roztaví v kelímku a taveninou se "navlhčí" povrch matrice. Safírové semeno se ponoří do taveniny na vrchol lisovací formy a vytáhne se vytvrzené do safíru ve tvaru žádané formy.

Metoda EFG umožňuje vytvářet různé tvary, které nejsou možné s jinými technologiemi, a proto šetří náklady spojené s obráběním a jinými dokončovacími procesy. Primární nevýhodou je doba a cena spojená s výrobou nástrojů k vytvoření tvarových krystalů. EFG-vyrobený krystalový materiál má typicky nízkou až střední optickou kvalitu a může být přesně vyráběn v různých krystalografických orientacích. Nejčastěji se používá pro mechanické, průmyslové a méně kvalitní optické aplikace [20].

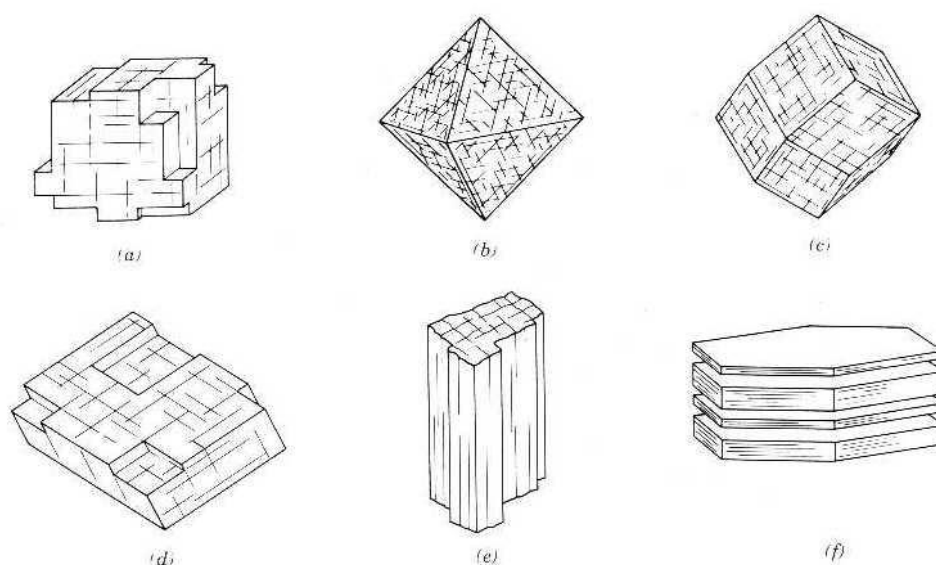
2.2.3 Štěpnost krystalu

Štěpnost je definována jako krystalograficky orientované minimum soudržnosti [25]. Jinými slovy je to schopnost krystalu štípat se nebo praskat při uderu podle určitých krystalografických směrů.

Štěpnost je podmíněna jedinečně vnitřní stavbou krystalu, proto je pro určitý minerál vždy stejná a nezávisí na vnějším tvaru krystalu. Štěpnost úzce souvisí s anizotropií vazebných sil, směr štěpnosti je závislý na nejslabších vazbách ve struktuře minerálu [14].

Pokud chceme štěpnost určitým způsobem definovat, musíme uvést její kvalitu a krystalografický směr. Směr udáváme buď Millerovým indexem nebo opisem jako např. kubická (100), oktaedrická (111), prizmatická (110) nebo pinakoidální (001) štěpnost atd. [14] (viz obr. 16).

- Kubické štěpení nastává, když existují tři roviny štěpení protínající se při 90 stupních (halit)
- Oktaedrické štěpení nastává, když jsou v krystalu čtyři štěpné roviny (fluorit, diamant)
- Dodekaedrické štěpení nastává, když je v krystalu šest štěpných rovin (Sphalerit)
- Prizmatické štěpení nastává, když existují dvě štěpné roviny v krystalu (spodumen)
- Pinakoidální štěpení nastává, když existuje jen jedna rovina štěpení (grafit) [26]

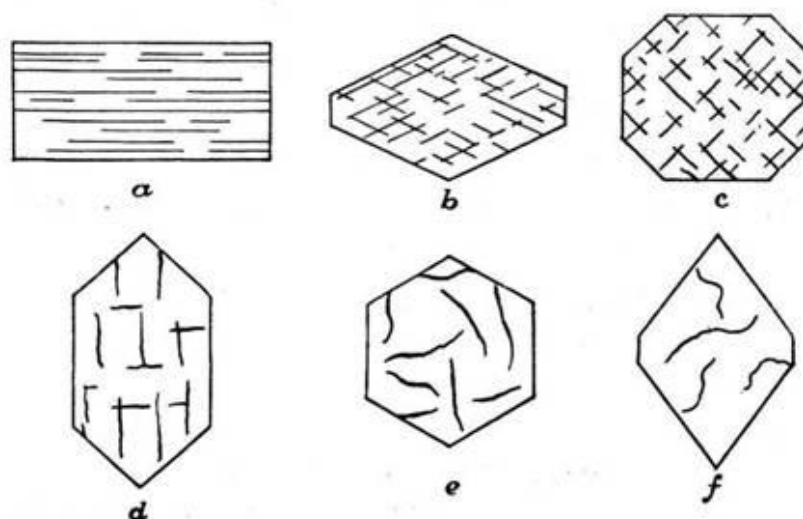


Obr. 16 Označení směru štěpnosti a) kubická, b) oktaedrická, c) dodekaedrická, d) klencová, e) prizmatická, f) pinakoidální (bazální) [26].

Kvalita štěpnosti se vyjadřuje nejčastěji v této škále:

- velmi dokonalá: krystal je skoro vždy schopný se dělit podle krystalových ploch, např. na šupinky (slídy, chlority)
- dokonalá: krystal se rozpadá převážně na úlomky ohraničené štěpnými plochami (kalcit, galenit)
- dobrá: na úlomcích krystalu se vyskytují jak zřetelně štěpné plochy, tak nerovné lomy náhodných lomů (živce)
- nedokonalá: většina ploch jsou plochy lomné, jen těžko a omezeně se projevují plošky štěpné (apatit, síra)
- velmi nedokonalá (chybějící): u těchto minerálů štěpnost prakticky chybí, projevuje se jen zcela výjimečně (korund, magnetit, zlato) [25]

Na Obr. 17 je znázorněna různá kvalita štěpnosti krystalů.



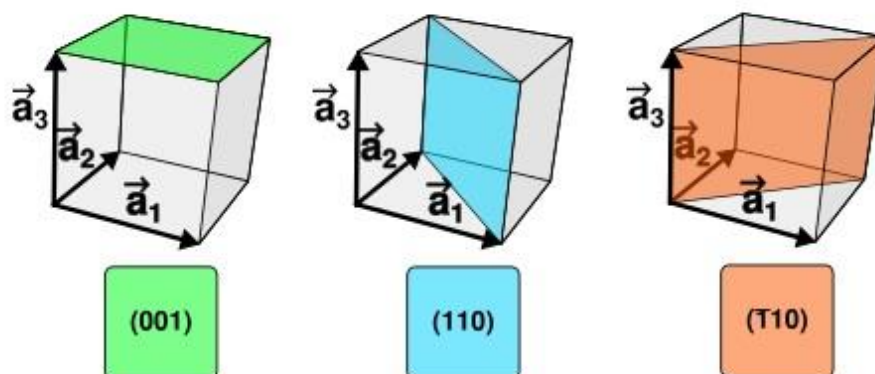
Obr. 17 Různá kvalita štěpnosti a) velmi dokonalá, b) dokonalá, c) dobrá, d) nedokonalá, e) špatná, f) chybějící [27]

Štěpnost velice omezuje možnost výřezu a výbrusu finálního výrobku (optického krystalu), proto je nutné k ní od začátku přihlížet a volit tvar směru výbrusu podle toho. V jiném případě hrozí zničení krystalu.

2.2.4 Millerovy indexy

Millerovy indexy (podle W. H. Millera) jsou indexy definující rovinu atomů v krystalu podle jejich průsečíků s krystalografickými osami tak, že se naleznou průsečíky roviny s třemi základními krystalografickými osami a označí se jako délky hran elementární buňky. Pak se reciproké hodnoty těchto veličin vydělí jejich největším společným

dělitelem tak, aby se dostala tři nejmenší možná čísla. Tyto čísla nám pak definují rovinu v krystalu (viz obr. 18).



Obr. 18 Významné Millerovy indexy [28]

2.3 Nástroje pro vývoj simulací

V této kapitole je podán přehled nástrojů potřebných pro tvorbu simulace.

2.3.1 Jazyk C++

Jazyk C++ je objektově orientovaný jazyk, citlivý na malá a velká písmena. „Proměnná“ a „proměnná“ označuje tedy jinou proměnnou. Identifikátor proměnné se může skládat z písmen, číslic a podtržítok, ale nesmí číslicí začínat. Přičemž bývá dobrým zvykem vyhradit podtržítka pro členské proměnné [29].

Deklarace jsou povinné. Deklaraci tvoří identifikátor a jméno. Podle umístění dělíme deklarace na globální a lokální. Jazyk obsahuje všechny standardní proměnné ve verzích se znaménkem i bez. Pomocí klíčového slova *const* vytváříme konstanty. Takto delegovanou proměnnou už nelze po přiřazení změnit.

Jazyk C++ má logický typ *bool*, ale vystačí si i jen s číselnou hodnotou, 0 pro *true* a nenulová pro *false*. Všechny příkazy musí být odděleny středníkem.

2.3.2 Výrazy a přiřazení

Jazyk C++ obsahuje celou řadu operandů a operátorů. Operátory dělíme podle arity na unární (tedy ty které reagují sami se sebou např. ++), binární (spojují dva výrazy, + a -) a ternární (např. operátor ?).

Základní příkazy jsou:

- Příkaz **if**, při splnění podmínky se provádí část programu za příkazem, při nesplnění se provádí větev **else**
- Příkaz **switch** v podstatě zjednodušuje příkaz if, v případě že je více možností, vnořováním příkazů by se stal kód nepřehledný. Klasicky se používá například při odchyování tlačítek klávesnice, každá klávesa pak provede jinou akci. Příkaz switch je specifický tím že každá část musí být ukončena příkazem **break**, jinak by program pokračoval dál, dalším příkazem
- Příkaz **while**, jde o základní smyčku, příkazy se provádějí tak dlouho dokud je splněna podmínka. V programech se často používá tzv. hrací smyčka (angl. gameloop), která se provádí tak dlouho dokud program běží, a program v ní vlastně „žije“

- Příkaz **do**, jde o verzi příkazu `while`, který se ale vždy provede aspoň jednou, tzn. že se podmínka kontroluje až po průchodu smyčky, a ne na začátku
- Příkaz **for** má nejčastěji podobu $(i = 0; i < n; i++)$ kde i je proměna cyklu a na začátku je jí přiřazena hodnota 0, opakováním cyklu se její hodnota zvyšuje o 1 tak dlouho dokud bude menší než n . Všechny tři parametry jsou nepovinné a mohou být vynechány (`;`) a k ukončení může být podobně použit příkaz `break`

Všechny tři smyčky jsou navzájem záměnné, jejich použití se řídí čitelností kódu.

2.3.3 Preprocesor

Preprocesor zpracovává zdrojový text programu před samotným překladem, vynechá komentáře a zavede záměnu textu (zamění jméno konstanty za číslo), také vkládá hlavičky souborů. Všechny příkazy pro preprocesor musí začínat znakem `#` a protože to nejsou příkazy jazyka C nejsou ukončeny středníkem. Mezi nejčastější příkazy preprocesoru patří `#ifndef`, `#define` a `#include`. Za každým příkazem následuje *ID*, za které preprocesor právě nahrazuje příslušnou hodnotu.

2.3.4 Funkce

Každá funkce musí mít definici a určené jméno kterým se volá. Volitelným parametrem jsou pak vstupní hodnoty a návratová hodnota. Pokud žádná návratová hodnota není, musí být uveden příkaz `void`. Protože se v jazyce C++ funkce píše do vlastního souboru, musí být funkce před použitím uvozena jejím prototypem, ten v podstatě překladači říká, co si má za neznámým příkazem představit.

Všechny funkce jsou implicitně deklarovány jako *extern*, to znamená že mohou být umístěny i do jiného než zdrojového souboru pomocí preprocesoru. A jsou pak v tomto souboru dostupná. Je-li ale deklarovaná ve třídě jako *static*, musí její definice následovat ve stejné překladové jednotce a je tedy definovaná pouze v ní. Pokud voláme funkci bez žádných vstupních parametrů musíme stejně uvést za jménem závorky, aby bylo jasné, že se jedná o volání funkce.

2.3.5 Ukazatel

Ukazatele jsou hlavním rozdílem, kterým se C++ liší od ostatních programovacích jazyků. Ukazatel jako takový je proměnná, která neukazuje na hodnotu, ale drží v sobě hodnotu adresy, na které je proměnná uložena v počítači. Na rozdíl od proměnné se ukazatel značí pomocí indikátoru, co je na adrese uloženo, operátoru hvězdičky * a názvu ukazatele. Ukazatel, se kterým chceme pracovat, musí být vždy inicializován na nějakou hodnotu a při vracení ukazatelů jako návratové hodnoty funkce si musíme dávat velký pozor, jestli je hodnota na adrese pořád platná, nebo jestli nebyla tato hodnota smazána při ukončení funkce a její hodnota nebyla přepsaná jinou, ve většině případu zcela náhodnou hodnotou. Tento problém řeší C++11 použitím „chytrých“ ukazatelů (angl. smart pointers) na úkor výkonnosti programu [29].

2.3.6 Knihovna GLM

OpenGL Mathematics je meziplatformová knihovna pro tvorbu grafického softwaru založeném na OpenGL Shading Language (GLSL) specifikaci. GLM poskytuje třídy a funkce a umožňuje práci s maticemi a vektory. Tento projekt není omezen jen na funkce GLSL. Systém poskytuje mnoho možností jako transformace matic, zapouzdření, náhodná čísla, rušení atd. [30].

2.3.7 Knihovna GLEW

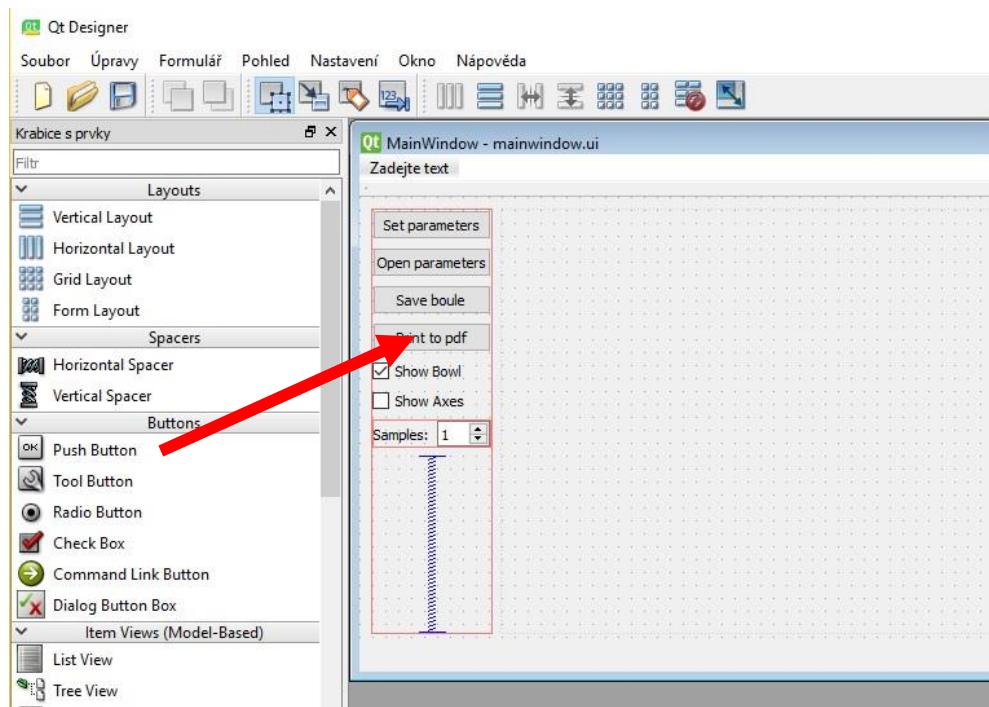
The OpenGL Extension Wrangler Library (GLEW) je meziplatformavé *open-source* C / C ++ rozšíření knihovny OpenGL. GLEW poskytuje účinné mechanismy při běhu programu pro určení, které rozšíření OpenGL jsou podporovány na cílové platformě. Funkce jádra OpenGL a jeho rozšířená funkčnost je nahraná v jediném hlavičkovém souboru. GLEW byl testován na různých operačních systémech, včetně systémů Windows, Linux, Mac OS X, FreeBSD, Irix a Solaris [31].

Nejprve je nutné vytvořit platný kontext renderování OpenGL a zavolat funkci *glewInit* pro inicializaci vstupních bodů tohoto rozšíření. Pokud *glewInit* vrátí GLEW_OK, inicializace byla úspěšná a může se použít dostupné rozšíření stejně jako základní funkce OpenGL.

2.3.8 Framework Qt

Qt je framework pro C++ vyvíjen finskou firmou Nokia a umožňuje snadno, intuitivně a rychle vytvářet uživatelské rozhraní pomocí umístění objektů do „widgetů“ [32].

Qt v základní verzi podporuje OpenGL a vytváření pdf. V základní nekomerční verzi je zdarma. Tento framework byl speciálně vyvinut pro rychlé tvoření uživatelských rozhraní a jeho ovládání na základní bási je vesměs intuitivní. Umožňuje snadné vytvoření kostry aplikace pomocí vytvoření svého interface, kde se z ovládacího panelu metodou „Drag and drop“, tedy táhnout a umístit, můžou natáhnout tlačítka, „spin boxy“ a widgety z panelu napravo a umístit na pracovní plochu uprostřed (viz obr. 19). QT pak sám vytvoří xml kód, který popisuje rozmístění a velikost jednotlivých prvků.



Obr. 19 Prostředí Qt designeru

Vyvíjet v Qt je možné jak v jeho vlastním prostředí (Qt creator), tak ho přímo integrovat do prostředí Visual Studia (VS) pomocí příslušného add-onu. Tato možnost sice neumožňuje pracovat s Qt tak pohodlně jako v jeho vlastním prostředí, ale zase přináší výhody VS. Já jsem se rozhodl pracovat ve VS, jeho prostředí je méně intuitivní ale nabízí více možností a není uživatele pracovat jen výhradně s nástroji Qt. Qt samotné už obsahuje podporu OpenGL, je ale potřeba vytvořit příslušný widget, vytvořit mu příslušnou třídu a v Qt povýšit příslušný widget k této třídě.

2.3.9 Knihovna OpenGL

Open Graphics Library je multijazykové, multiplatformové aplikační programovatelné rozhraní (API) na renderování 2D a 3D vektorové grafiky. Toto rozhraní pracuje hlavně s grafickým procesorem (GPU) a dosahuje tak hardwarem urychlené renderování, tedy renderování, které bude vždy rychlejší než softwarové vykreslení. OpenGL je spravováno neziskovou organizací The Khronos Group [33].

OpenGL také definuje vlastní proměnné, aby byla zajištěna přenositelnost na jinou platformu (např. Mac). V případě Windows se pak jedná jen o přetypování původních proměnných. Názvy nových proměnných se výrazně podobají těm původním, ale navíc obsahují předponu GL (*GLfloat*, *GLuint*, *GL_FALSE*). Na to je nutno dávat pozor, aby nedocházelo k automatickému přetypování a následné ztrátě dat nebo přesnosti.

Každá třída založená na OpenGL musí zdědit minimálně tři funkce *initializeGL*, tato třída obsahuje všechny parametry nutné k načtení prázdného okna a jak název napovídá inicializuje všechny povinné parametry a spouští všechny podpůrné funkce jako je vyhlazování hran a podobně. Tuto funkci volá sám Qt na začátku chodu programu.

Funkce *paintGL* obstarává samotná vykreslení. Jsou v ní umístěny všechny vykreslovací příkazy. Je nutné je v programu volat pořád dokola a změnou parametrů vykreslení vzniká efekt pohybu.

Funkce *ResizeGL* je zodpovědná za správné zachování poměru stran zobrazení, je volána při změně poměru stran vlastního okna programu.

Pro správnou funkčnost každého zobrazení jsou nutné tři matice.

- Matice Model (Model Matrix)
Hýbe, natáčí a zvětšuje (zmenšuje) jednotlivé objekty
- Matice pohledu (View Matrix)
Hýbe celým světem, a tím nastavuje polohu kamery
- Matice projekce (Projection matrix)
Vypočítává zobrazení světa, tvoří perspektivu nebo naopak „efekt menu“, tedy 2D grafika na 3D pozadí

Pomocí těchto tří matic se dopočítává poloha každého bodu jak v případě 2D zobrazení, tak i spočítat zobrazení způsobené perspektivou 3D. OpenGL samotné

neobsahuje žádnou podporu písma ani jiného druhu zobrazení znaků. Ve své specifikaci podporuje jen 3 základní tvary (bod, úsečku a trojúhelník), z těchto nástrojů jsme schopni vymodelovat jakýkoliv jiný tvar.

Color Buffer

Vykreslování v OpenGL funguje na jednoduchém principu zápisu barvy pixelu do Color Bufferu. Tam je po každém průchodu systému zapsána každá vypočítaná barva z grafické karty ze zadaných dat interpolací.

Depth Buffer

V základním režimu zobrazuje OpenGL všechna data podle pořadí, ve kterém byli do grafické karty posláni. To znamená, že na místo pixelu zapíše první barvu a při následném zapisování na stejnou pozici jí bez rozmyslu přepíše.

Po zapnutí Depth Bufferu (příkaz *glEnable(GL_DEPTH_TEST)*) začne OpenGL pro každý ukládat nejen barvu ale i hloubku do zvláštního bufferu a před každým zapsáním zkontroluje číslo na této pozici. Pokud je číslo větší, zápis se zruší. Pokud je číslo menší, barva se zapíše a zapíše se i nová hodnota do Depth Bufferu. Pro správnou funkci je nutné před každým průchodem programu tento buffer vymazat (příkaz *glClear(GL_DEPTH_BUFFER_BIT)*).

Dva režimy

V současné době funguje OpenGL ve dvou režimech zadávání dat. V původním omezeném, intuitivnějším, ale také pomalejším na vykreslení. Jde o princip, kde je zobrazovací kód OpenGL psán do bloků mezi značky *glBegin(značka příznaku vykreslování)* a *glEnd()*.

Druhý modernější postup k datům přistupuje pomocí *pipeline*. Tyta data jsou pak spravována přímo na grafické kartě pomocí mikroprogramu (shaderu), zde se zpracovává jak poloha (*vertex*) tak se dopočítává i barva celého tvaru (*fragment*).

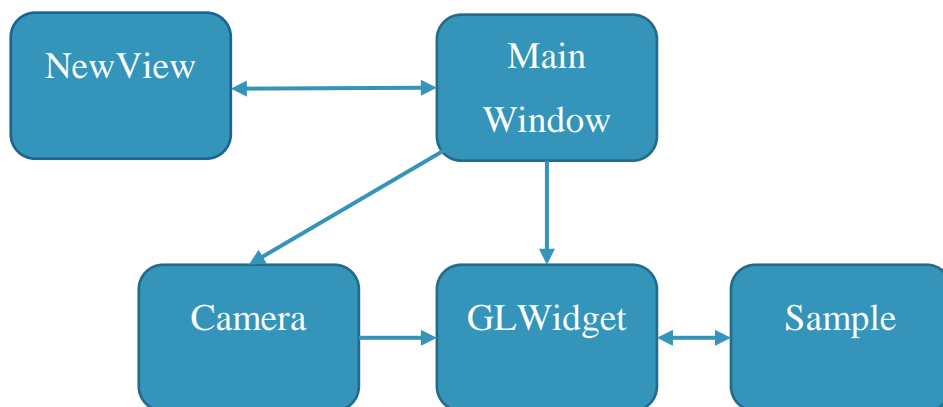
3. Návrh a realizace programu

Cílem praktické části diplomové práce bylo získat poznatky o programování v C++ a využít je k vytvoření programu, který by umožňoval vizualizaci a zároveň by našel optimální řešení výřezu krystalu, tak aby byl co nejmenší odpad materiálu.

Zobrazující část programu zajišťuje OpenGL popsány v kapitole (2.3.9). Program samotný zajišťuje framework Qt (2.3.8). K implementaci bylo použito vývojové prostředí Visual Studio 2015.

3.1 Postup vývoje v jazyce C++

První krok při výrobě každého softwaru je návrh. Dle mého návrhu se třída *MainWindow* zabývá všemi vstupy a výstupy, navíc je předává zobrazovací třídě *GLWidget*. Tato třída dále vlastní třídu *Camera* a *Sample*. První jmenovaná třída má za úkol imitovat kameru. Kamera jako taková v OpenGL neexistuje, ale podobného efektu se můžeme přiblížit, když celou scénu posuneme v opačném směru. Celého procesu se dosahuje pomocí maticového počtu. Pro ulehčení práce s maticemi slouží knihovna GLM (2.3.6)



Obr. 20 Diagramy tříd užitých v aplikaci

3.1.1 Třída MainWindow

Třída MainWindow tvoří kostru celého programu, dědí všechny funkce QMainWindow. Jsou v ní obsaženy procedury pro zpracování dat zadané uživatelem, tak pro čtení dat ze souboru

3.1.2 Třída GLWidget

Ve třídě se nacházejí funkce odpovědné za zobrazovací logiku. Mezi nejdůležitější z nich patří funkce *isSampleValid* odpovědná za barvu samplu.

```
bool Sample::isSampleValid(const std::vector<float>& data)
```

Z prototypu funkce je vidět, že Sample dostane na vstup popis krystalu a do návratové hodnoty sám rozhodne, jestli neleží uvnitř nebo ne, tomu pak odpovídá i barva.

3.1.3 Třída Sample

Třída Sample zajišťuje veškerou práci s výsledným vzorkem. Umožňuje jeho posun, změnu umístění a v poslední radě počítá i jeho optimální umístění v krystalu (bouli).

Mezi nejdůležitější funkce třídy patří *countSample*, *findFit* a *biggest*. Poslední jmenovaná funkce zajišťuje, vytvořený Sample byl podle zadaných parametrů co největší, a ne přitom pořad ležel uvnitř zadané boule. Dociluje toho tak, že se pořad snaží vytvořit větší Sample a u něj zjišťuje, jestli je pořad platný, tedy jestli jeho všechny body leží vně po odečtení zadané rezervy. Pokud je to pravda předá ho zpátky widgetu k vykreslení viz Obr. 21.

1	WHILE(true)
2	Zavolání funkce zvětšení Samplu
3	Posune Sample tak, aby byl, jak jen to je možný blízko nule
4	Zarovná Sample na střed osy válce
5	IF (Sample je uvnitř boule)
6	CONTINUE
7	ELSE
8	Zapiše se poslední validní stav a funkce se ukončí

Obr. 21 Algoritmus funkce biggest

Funkce *findFit*, u již vytvořeného *Samplu* hlídá, jestli jeho poloha co nejvíce vyhovuje, to znamená, že posouvá *Sample* tím směrem v bouli tak, aby funkce *biggest* mohla optimálně plnit svoje zadání.

countSample, tato funkce pak z dat získaných pomocí předchozích sestaví na v hodném místě *Sample*, spočítá polohu bodu jednotlivých vrcholu a předá ho je OpenGL k zobrazení. Všechny tyto funkce se spouští po zadání nebo načtení dat uživatelem. K tomuto účelu využívá hlavně knihovnu GLM, která obsahuje příslušné matice pro posun rotaci a zvětšení

```
void Sample::countSample()
{
    glm::mat4 translate = glm::translate(glm::mat4(1.f), -_position);
    glm::vec3 scalator{ _scaleX, _scaleY, _scaleZ };
    glm::mat4 scale = glm::scale(scalator);

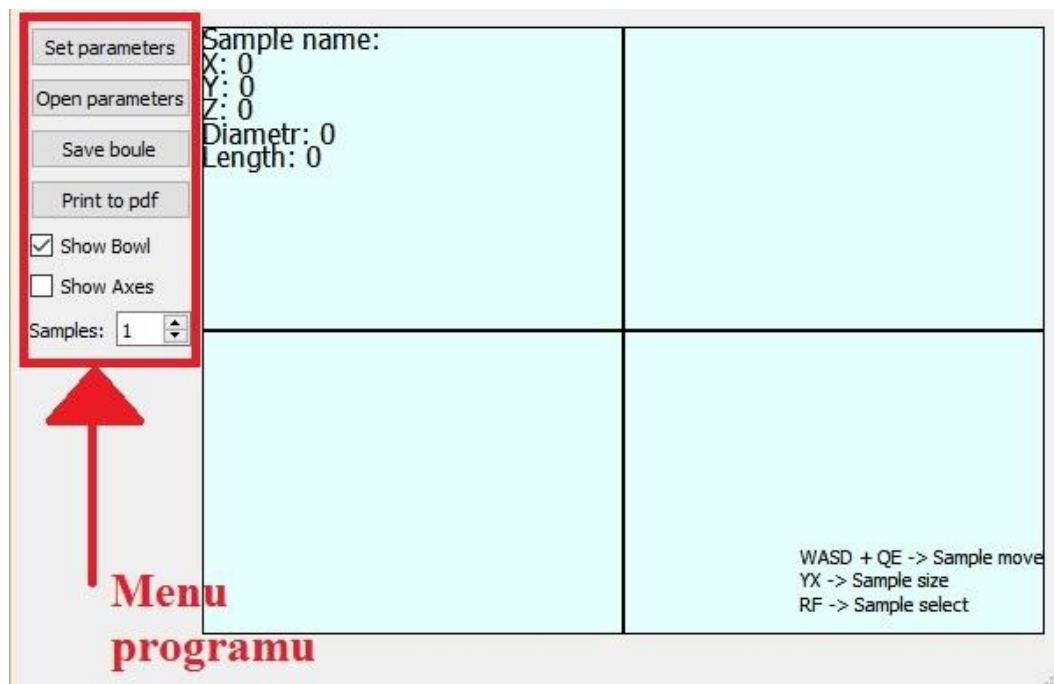
    glm::mat4 rotator = translate*
        glm::rotate(_angle1, glm::vec3{ 1.f, 0.f, 0.f })* //Xaxe
        glm::rotate(_angle2, glm::vec3{ 0.f, 1.f, 0.f })* //Yaxe
        glm::rotate(_angle3, glm::vec3{ 0.f, 0.f, 1.f })*scale; //Zaxe

    this->Vertexs[
A] = rotator*glm::vec4{ -0.5f, +0.5f, +1.0f , +1.f };
    this->Vertexs[B] = rotator*glm::vec4{ -0.5f, -0.5f, +1.0f , +1.f };
    this->Vertexs[C] = rotator*glm::vec4{ +0.5f, -0.5f, +1.0f , +1.f };
    this->Vertexs[D] = rotator*glm::vec4{ +0.5f, +0.5f, +1.0f , +1.f };
    this->Vertexs[E] = rotator*glm::vec4{ -0.5f, +0.5f, +0.0f , +1.f };
    this->Vertexs[F] = rotator*glm::vec4{ -0.5f, -0.5f, +0.0f , +1.f };
    this->Vertexs[G] = rotator*glm::vec4{ +0.5f, -0.5f, +0.0f , +1.f };
    this->Vertexs[H] = rotator*glm::vec4{ +0.5f, +0.5f, +0.0f , +1.f };
}
}
```

Obr. 22 Ukázka kódu pro výpočet *Samplu*

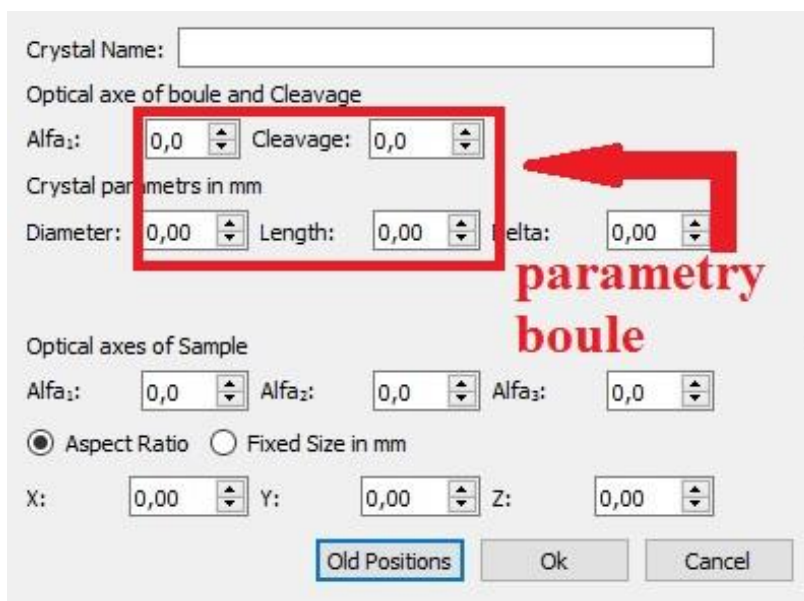
3.2 Popis funkce programu

Po spuštění programu samotného se uživatel dostane do hlavního okna programu. Většinu programu zaujímá zobrazovací okno rozdělené do čtyř kvadrantů, nárys, bokorys, půdorys a celkový 3D pohled, tak jak to bývá zvykem u výkresových nástrojů typu CAD.



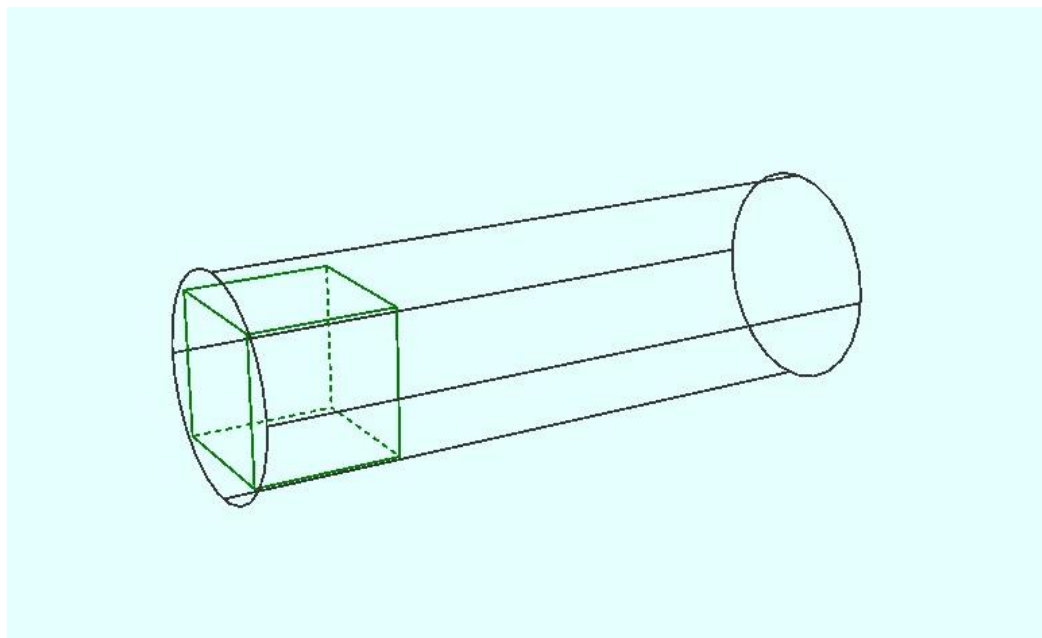
Obr. 23 Hlavní menu programu

V levé části programu je ovládací menu skládající se ze čtyř tlačítek (viz Obr. 24). Po zmáčknutí prvního z nich se otevře nové menu, do které ho uživatel musí zadat parametry krystalu a také parametry samplu. Uživateli jsou k dispozici tři možnosti, jak menu opustit. Možnost zrušit neprovede nic, možnost ok vypočítá a zobrazí samplu s vhodným umístěním, a možnost „Old Positions“ dává uživateli možnost přepočítat samplu podle již dříve umístěných poloh.



Obr. 24 Nastavení parametru boule

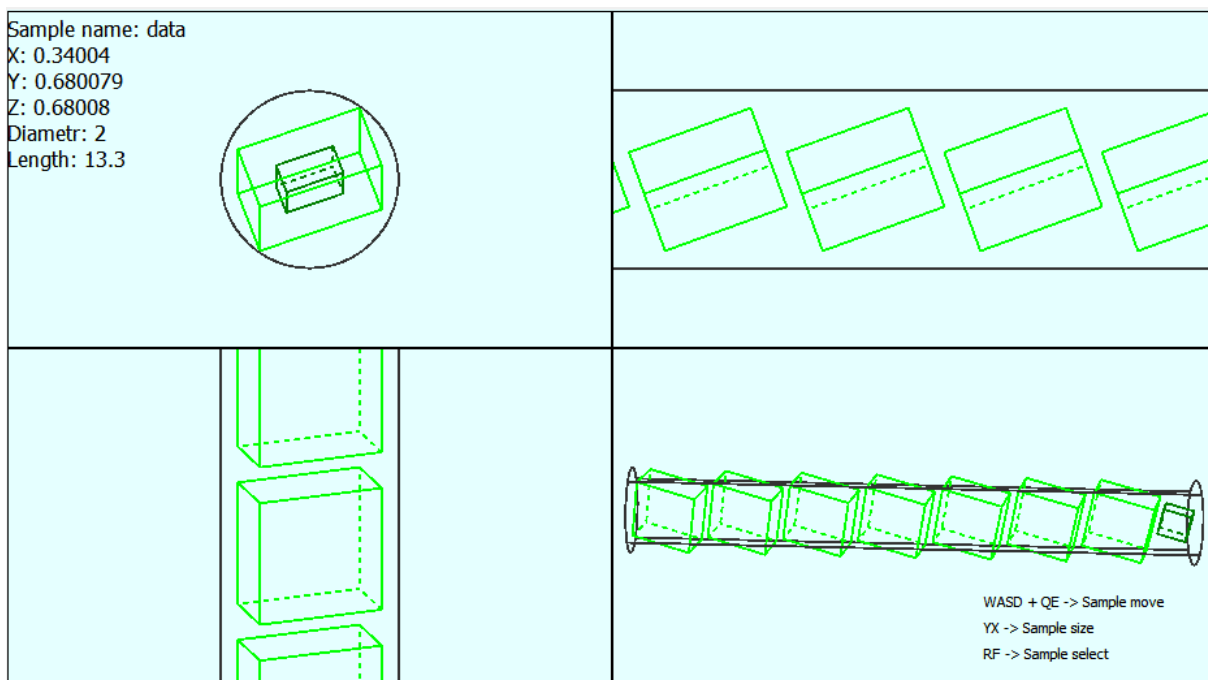
V hlavním zobrazení se uživatel může pohybovat myší (posouvat pohledy, zoom) nebo pomocí klávesových zkratk. Klávesy R a F dávají uživateli vybrat sample se kterým právě hodlá pracovat. Takový to vzorek je pak barevně od ostatních rozlišen. Uživatel má možnost pomocí WASD hýbat samplem v ose x a y, pomocí QE v ose z. pomocí klávesy Y a X má uživatel možnost měnit skokově velikost vzorku.



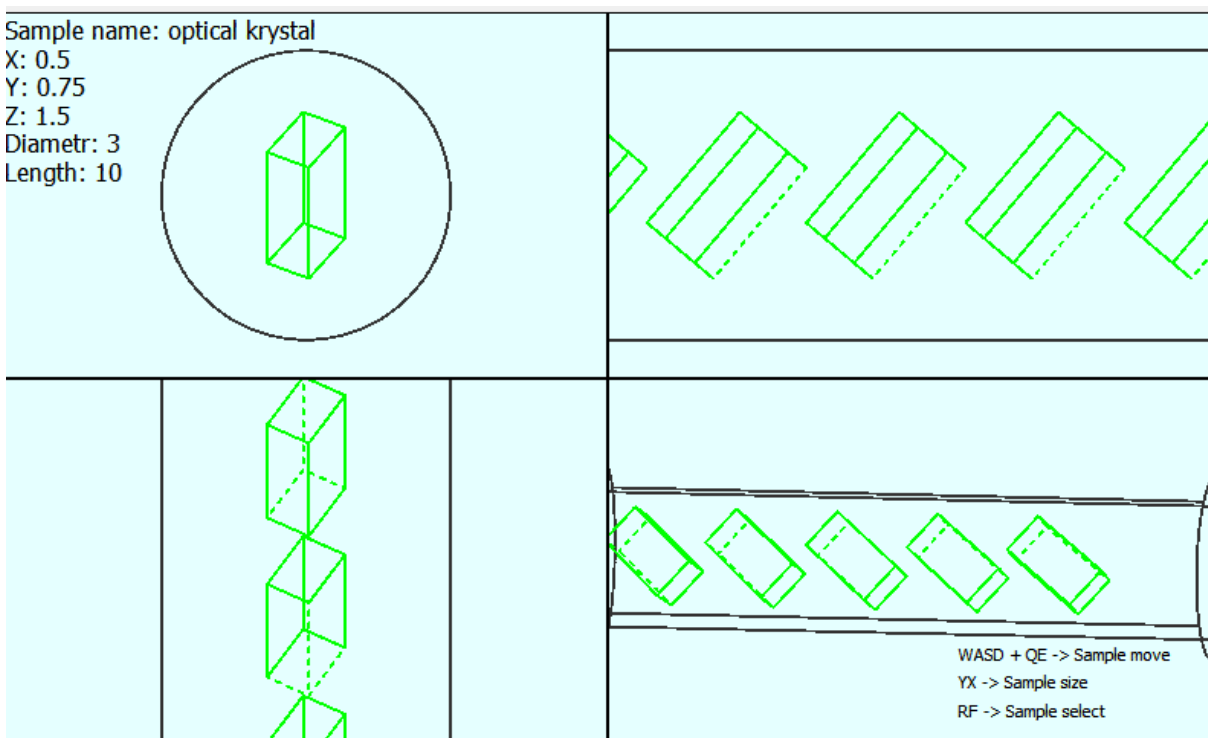
Obr. 25 Náhled celé boule za použití perspektivy

Program umožňuje dva režimy (viz Obr. 24) zobrazení, pomocí zadání poměru stran a nalezení největšího samplu (viz Obr. 25) nebo fixním zadáním stran (Obr 26)

Dále je přítomno tlačítko na nahrání dříve rozdělané a uložené práce, v neposlední řadě možnost vygenerovat obrázkové pdf s parametry výsledné práce.



Obr. 26 Ukázka optimálních výřezů krystalu o maximální velikosti



Obr. 27 Ukázka výřezu krystalu s danou velikostí

4. Závěr

Tato práce si kladla za cíl navrhnout program, který by simuloval řez broulí. Prvním krokem pro dosažení tohoto cíle bylo seznámení se s vlastnostmi světla při průchodu anizotropním prostředím. Na základě těchto znalostí a s využitím jazyka c++ byl vytvořen program s grafickým rozhraním OpenGL. Ten umožňuje uživateli zobrazit celý krystal a vněm spočítat umístění jednotlivých optimálních řezů za využití vyhledávacích algoritmu pomocí numerického řešení.

Aplikace umožňuje vložit až 10 výřezu (sample), ty se podle zadání umístí na optimální pozice. Pokud se výřez do krystalu nevejde, je na to uživatel barevně upozorněn a má možnost si výřezy přeskládat dle svého uvážení (např. zmenšit a přemístit). Aplikace správně reaguje na všechna natočení optických os krystalu.

Velkým přínosem toho řešení je, že zprostředkování vizuální informace uživateli a tím snižuje nároky na jeho představivost a umožňuje mu efektivněji využít tento drahý materiál. Tím dochází k větší výtěžnosti a ušetření nemalých výrobních nákladů.

Výhodou programu je možnost exportu do pdf. To zaručuje snadné a přesné předání parametrů řezu a výsledného krystalu od navrhovatele k řezači krystalu.

4.1 Možnosti rozšíření

Práce umožňuje velké množství rozšíření, jako je přidání možnosti zobrazení průchodu světla výřezem, přidání možnosti změny tvaru výřezu.

5. Zdroje

1. *VISION* by Pierre Buser and Michel Imbert (MIT Press, Cambridge, MA, 1992). SIGART Bull., 1993.
2. http://www.vascak.cz/data/android/physicsatschool/galerie/big_opt_vlna.png. 2017.
3. Saleh, B.E.A. and M.C. Teich, *Základy fotoniky, svazek 1*. Praha: MATFYZPRESS.
4. http://fyzika.fyzikaprozivot.cz/multimedia/photo/7-rocnik/5_3/lom-svetla.png. 2017.
5. Bennett, C.A., *Principles of Physical Optics*. 2008: Wiley.
6. Halliday, D., Resnick, R. & Walker, J., *Fyzika: vysokoškolská učebnice obecné fyziky*. 2000, Brno: VUTIUM
7. Vrbová, M., *Lasery a moderní optika*. 1. vyd. ed. oborová encyklopedie. 1994, Praha: Prometheus. 474.
8. http://i.idnes.cz/07/103/cl/JLB1ea118_polarizace.jpg. 2017.
9. Kalus, R., *Trivium z optiky*. 2004, Ostrava: Ostravská univerzita.
10. Saleh, B.E.A. and M.C. Teich, *Základy fotoniky, svazek 2*. Praha: MATFYZPRESS.
11. *how-to-design-birefringent-polarizers*. 2017; Available from: <http://www.zemax.com/os/resources/learn/knowledgebase/how-to-design-birefringent-polarizers>.
12. *Světlo v anizotropním prostředí – optika krystalů*. 2017.
13. <http://webphysics.davidson.edu>. 2017.
14. Václav Vávra, Z.L. *Multimediální studijní texty z mineralogie pro bakalářské studium*. Available from: <http://mineralogie.sci.muni.cz/index.htm>.
15. J., R. and V. M. *Encyklopedie fyziky*. Available from: <http://fyzika.jreichl.com>.
16. *obecné mineralogie*. Available from: http://muzeum.mineral.cz/mineralogie/ucebnice/obecna_min/.
17. *Experimentální metody pevných látek*.
18. Růžička, M., *Škola růstu krystalů 2000 METODY RŮSTU KRYSTALŮ* Bratislava: Chemickotechnologická fakulta STU.
19. *kde-se-bere-monokrystalicky-kremik*. Available from: kde-se-bere-monokrystalicky-kremik.html.
20. *Growth*. 2017; Available from: <http://www.clearlyapphire.com/Growth.html>.
21. krystalů, Č.s.p.r., *Technologie růstu monokrystalů křemíku Czochralskiho metodou*. 2004.
22. <http://www.gtat.com/Collateral/Images/English-US/Sapphire%20Material/Optical%20Sapphire%20Boule.jpg>. 2017.
23. Nitsch, K., *RŮST KRYSTALŮ BRIDGMANOVOU METODOU*
24. <http://www.bagdasarovcrystals.com/v1/index.php?id=1>. 2017.
25. *Úvod do mineralogie*. Available from: <http://www-ldf.mendelu.cz/ustavy/geologie/studium.htm>.
26. Hurlbut, C.S.K., Cornelis, , *Manual of Mineralog*. 1985: Wiley.
27. Konta, H.a., *Horninotvorné minerály*. 1959.
28. https://cs.wikipedia.org/wiki/Millerovy_indexy#/media/File:Miller_Indices_Felix_Kling.svg. 2017.
29. Jiří, P., *Algoritmy v jazyku C a C++: 3., aktualizované a rozšířené vydán*. 2015: Grada Publishing a.s.
30. *GLM*. 2017; Available from: <http://glm.g-truc.net/0.9.8/index.html>.
31. *GLEW*. Available from: <http://glew.sourceforge.net/>.
32. *Qt*. 2017 [cited 2017; Available from: <https://www.qt.io/>].
33. *OpenGL*. 2017 [cited 2017; Available from: <https://www.opengl.org/>].

6. Přílohy

6.1 Knihovny nutné pro spuštění programu

- glew32.dll
- icudt54.dll
- icuin54.dll
- icuuc54.dll
- libEGL.dll
- libGLESv2
- libwinpthread-1.dll
- msvcp140.dll
- Qt5Core.dll
- Qt5Gui.dll
- Qt5OpenGL.dll
- Qt5PrintSupport.dll
- Qt5Widgets.dll
- qwindows.dll