

České vysoké učení technické v Praze

Fakulta elektrotechnická

Katedra elektrotechnologie



Optický model rozptylu světla využívající  
modifikovaný koherentní maticový formalismus

Optical model of light scattering using modified coherent matrix  
formalism

diplomová práce

Radek Nevyhoštěný

**Studijní program:** Elektrotechnika, energetika a management

**Studijní obor:** Technologické systémy

**Vedoucí práce:** doc. Mgr. Jakub Holovský, Ph.D.

Praha 2018



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Nevyhoštěný** Jméno: **Radek** Osobní číslo: **420023**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra elektrotechnologie**  
Studijní program: **Elektrotechnika, energetika a management**  
Studijní obor: **Technologické systémy**

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Optický model rozptylu světla využívající modifikovaný koherentní maticový formalismus**

Název diplomové práce anglicky:

**Optical model of light scattering usány modified coherent matrix formalism**

Pokyny pro vypracování:

1. Model šíření světla ve struktuře s hladkými rozhraními pomocí koherentního formalismu
2. Grafické rozhraní pro návrh struktury, zadávání parametrů a vykreslování výsledků
3. Rozšíření koherentních matic pro výpočet více úhlů současně
4. Rozdělení na koherentně se šířící intenzitu a nekoherentně se šířící intenzitu
5. Zahmutí rozptylu pomocí nedíagonálních členů rozptylových matic
6. Zahmutí manipulace fáze do nekoherentní části
7. Testování, srovnání s dostupnými výpočetními postupy

Seznam doporučené literatury:

- [1] článek: R. Santbergen, et al., Solar Energy Materials & Solar Cells 92 (2008) 432?444
- [2] článek: M. C. Tropicovsky, et al., OPTICS EXPRESS, vol 18 (2010) p 24715
- [3] kniha: R. B. Wehrspohn, U. Rau: Photon, Management in Solar Cells, ISBN: 978-3527411757
- [4] <https://forms.ise.fraunhofer.de/en/optos>

Jméno a pracoviště vedoucí(ho) diplomové práce:

**Mgr. Jakub Holovský, Ph.D., katedra elektrotechnologie FEL**

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

\_\_\_\_\_

Datum zadání diplomové práce: **15.02.2018**

Termín odevzdání diplomové práce: \_\_\_\_\_

Platnost zadání diplomové práce: **30.09.2019**

\_\_\_\_\_  
Mgr. Jakub Holovský, Ph.D.  
podpis vedoucí(ho) práce

\_\_\_\_\_  
podpis vedoucí(ho) ústavu/katedry

\_\_\_\_\_  
prof. Ing. Pavel Ripka, CSc.  
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



## 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.

V Praze, dne 24.5.2018

.....

Radek Nevyhoštěný



## Poděkování

Rád bych poděkoval vedoucímu své diplomové práce panu doc. Mgr. Jakubu Holovskému, Ph.D. za odborné vedení a cenné rady při zpracování této práce.





**Název práce:** Optický model rozptylu světla využívající modifikovaný koherentní maticový formalismus

**Autor:** Radek Nevyhoštěný

**Vedoucí práce:** doc. Mgr. Jakub Holovský, Ph.D.

**Abstrakt:** Tato práce se zabývá vývojem programu pro simulaci chování světla ve vícevrstvých strukturách, potenciálně umožňující zahrnutí difuze světla. V teoretické části je odvozen matematický model popisu šíření světla a jeho difuze pomocí modifikovaného maticového formalismu. Dále je zde popsán výběr programového jazyka pro vývoj a princip chodu využitých externích knihoven.

Praktickým výstupem práce je program realizovaný v jazyce Python, který umožňuje návrh a vizualizaci struktury materiálů přes grafické uživatelské rozhraní. Pro takto navrženou strukturu je možné simulovat průběh reflexe, transmise a absorpce pro obě polarizace světla. Další funkcionalitou je možnost umístění difuzních rovin do struktury pro simulaci difuze světla ve struktuře. Pro tento případ je v praktické části odvozeno rozšíření maticového výpočtu do diskretizovaného úhlového vektoru a způsob vytvoření difuzní matice. Počet úhlů, pro které probíhá simulace, je uživatelsky nastavitelný spolu s rozsahem výpočtu. Výsledná matice je programem převáděna na soustavu lineárních rovnic a řešena. Výsledky simulací lze exportovat do standardních grafických a textových formátů.

V závěru práce je program porovnán s konkurenčními simulačními programy OPAL 2 od PV Lighthouse a OPTOS od instituce Fraunhofer ISE.

**Klíčová slova:** rozptyl světla, maticový model, python, koherentní formalismus

**Title:** Optical model of light scattering using modified coherent matrix formalism

**Author:** Radek Nevyhoštěný

**Supervisor:** doc. Mgr. Jakub Holovský, Ph.D.

**Abstract:** The aim of this work is the development of a software for simulation of the light propagation in a multilayer structures, potentially allowing inclusion of diffusion of light. In the theoretical part of the work, mathematical model of light propagation using modified matrix formalism is derived along with diffusion of such light. Furthermore, the process of choosing programming language for the development and the internal workings of the external libraries used is described in this part.

Practical output of the work is standalone software realized in the Python programming language, which allows the user to design and visualize a structure of multilayer material through the graphical user interface. For the designed structure reflection, transmission and absorption can be simulated for both polarizations of light. Another functionality is the option to insert a diffusion plane into the structure for simulation of a light diffusion in the structure. For this purpose, the matrix mathematical model is extended into a discrete angle vector and method of the diffusion matrix creation is described. Number of the angles, for which the simulation is computed, is set by the user alongside the range of the simulation. Results of the simulation can be exported to the standard graphics and text formats.

In the final part of the work developed software is compared to existing solutions OPAL 2 from PV Lighthouse and OPTOS from Fraunhofer ISE institute.

**Keywords:** diffusion of light, matrix model, python, coherent formalism

# Obsah

<b>Seznam symbolů a zkratk</b> .....	<b>13</b>
<b>1 Úvod</b> .....	<b>15</b>
<b>2 Matematický popis šíření světla v tenkých vrstvách</b> .....	<b>17</b>
2.1 Rovnice vlny .....	17
2.2 Vlna na rozhraní .....	17
2.3 Fresnelovy rovnice.....	19
2.4 Intenzita reflexe a transmise.....	22
2.5 Změna fáze na rozhraní .....	22
2.6 Brewsterův úhel .....	23
2.7 Jednovrstvá struktura.....	24
2.8 Mnohovrstvá struktura .....	25
<b>3 Programovací jazyk Python</b> .....	<b>33</b>
3.1 Historie .....	33
3.2 Funkcionalita .....	34
3.2.1 Datové typy a proměnné.....	34
3.3 Knihovny jazyka Python .....	36
3.3.1 Knihovna NumPy.....	36
3.3.2 Knihovna SciPy.....	38
3.3.3 Knihovna Matplotlib.....	38
3.3.4 Knihovna Tkinter .....	38
<b>4 Vývoj samotného programu RLay</b> .....	<b>39</b>
4.1 Struktura programu.....	39
4.2 Modul Mainframe.....	40
4.3 Modul DataManagment.....	43
4.4 Modul Calculation .....	45
4.4.1 Polarizace vlny .....	45
4.4.2 Úhlový vektor .....	46
4.4.3 Difuze ve struktuře .....	48
4.4.4 Výsledná rozptylová matice .....	50
4.4.5 Řešení rozptylové matice .....	52
<b>5 Porovnání s existujícími programy</b> .....	<b>57</b>
5.1 PV Lighthouse .....	57
5.2 Fraunhofer ISE OPTOS .....	59
<b>6 Porovnání s experimentálními daty</b> .....	<b>61</b>

<b>7 Závěr .....</b>	<b>63</b>
<b>Použité zdroje .....</b>	<b>65</b>
<b>Seznam obrázků .....</b>	<b>67</b>
<b>Seznam ukázek kódu .....</b>	<b>68</b>
<b>Seznam tabulek .....</b>	<b>68</b>

# Seznam symbolů a zkratek

$A$  - matice systému

$ADF()$  - úhlové rozložení

$b$  - vektor pravé strany

$B$  - magnetická indukce

$\beta$  - fázový posun vlny

$c$  - rychlost světla ve vakuu

$d$  - tloušťka materiálu

$D$  - elektrická indukce

$D$  - difuzní matice

$\Delta$  - rozdíl optické dráhy

$E$  - amplituda intenzity elektrického pole

$\vec{E}$  - vektor intenzity elektrického pole

$\varepsilon$  - permitivita

$\varepsilon^{+,-}$  - komplexní amplituda intenzity elektrického pole vlny

$e$  - Eulerovo číslo

$H$  - amplituda intenzity magnetického pole

$\vec{H}$  - vektor intenzity magnetického pole

$i$  - imaginární jednotka

$I$  - matice rozhraní

$I$  - intenzita vlny

$J$  - proudová hustota

$k$  - vlnové číslo

$\vec{k}$  - vlnový vektor

$\kappa$  - extinkční koeficient

$\lambda$  - vlnová délka

$L$  - matice šíření ve vrstvě

$n$  - reálná část indexu lomu

$N$  - komplexní index lomu

$M$  - počet vrstev struktury

$r$  - reflexní koeficient

$R$  - intenzita reflexe

$S$  - rozptylová matice

$s()$  - difuzní faktor

$t$  - transmisní koeficient

$T$  - intenzita transmise

$\mu$  - permeabilita

$\theta$  - úhel

$\theta_B$  - Brewsterův úhel

$x$  - osa  $x$  souřadnicového systému

$y$  - osa  $y$  souřadnicového systému

$z$  - osa  $z$  souřadnicového systému

# 1 Úvod

Spolu s rozvojem solárních článků v posledním desetiletí roste i produkce a vývoj tenkých vrstev optického charakteru. Od roku 2012 celosvětový průmysl v tomto odvětví zaznamenává stabilní růst přes 15 % a výhledově je produkce odhadována na více než 600 kilotun ročně pro rok 2020. Zdokonalení technologií vytváření těchto vrstev spolu s sebou nese nové možnosti depozice složitých multivrstev za účelem dosažení požadované optické odezvy výsledné struktury. Schopnost tyto struktury efektivně modelovat je důležitou součástí stávajícího vývoje. [1]

Cílem této práce je navržení matematického modelu rozptylu světla ve struktuře využívajícího maticový formalismus a jeho programová realizace. V často používaném typu výpočtu interakce světla se strukturou je postupováno metodou sledování paprsku (*Ray tracing*), kdy je pro dopadající světelnou vlnu postupně počítána množina všech cest světelné vlny strukturou, na kterou dopadá. Při každé interakci se paprsek zeslabí a přesnost tohoto výpočtu je dána úrovní zeslabení, do které je paprsek sledován. Oproti tomu v maticové metodě jde v principu o analytické řešení, které řeší nekonečné množství interakcí. V tomto postupu se nejprve vytvoří matice svazující světelné vlny na obou stranách struktury. Tato matice charakterizuje danou strukturu jako celek a popisuje nejen všechny interakce, ke kterým dochází na rozhraní struktury s okolním prostředím, ale i veškeré interakce uvnitř této struktury. Výhodou tohoto postupu je značné ušetření výpočetního času potřebného k modelování chování světla v dané struktuře.

Hlavním výstupem této práce je program umožňující návrh libovolné tenkovrstvé struktury a simulaci jejího chování při interakci se světlem. Na rozdíl od základního maticového modelování struktury je úkolem navrhnout modifikaci tohoto výpočtu pro zahrnutí difuze světla uvnitř struktury. Vlivem difuze světla dochází k odchýlení části paprsku do jiných směrů šíření. Výsledný rozměr svazujících matic narůstá s počtem úhlů, do kterých je světlo difundováno. Vyvíjený program musí být proto schopen řešit úkony prováděné nad rozměrnými maticemi často komplexního charakteru. Z tohoto důvodu je pro vývoj zvolen jazyk Python. Jednou z jeho hlavních předností je distribuce pod open-source licencí a množství volně využitelných externích knihoven. Pro vědecké výpočty je dostupná knihovna NumPy, která je už ve svém základu založena na počítání s vícedimenzionálními poli. Pro jednoduchost obsluhy bude program doplněn grafickým uživatelským rozhraním a modulem pro vykreslení výsledných simulací s možností exportu do textových a grafických formátů.

V závěru práce bude vyvinutý porovnán s dostupnými programy podobného zaměření. Výstup simulace bude také porovnán s experimentálně získanými daty reflexe solárních článků.





## 2 Matematický popis šíření světla v tenkých vrstvách

V této kapitole budou popsány základní matematické principy použité k analýze chování světla ve vícevrstvých strukturách. Z hlediska zaměření práce jsou vztahy odvozovány pouze pro hladká rozhraní.

### 2.1 Rovnice vlny

Z Maxwellových rovnic lze pro obecnou rovinnou vlnu elektrického pole  $\vec{E}$  odvodit vztah:

$$\frac{\partial^2 \vec{E}}{\partial x^2} - \mu\epsilon \frac{\partial^2 \vec{E}}{\partial t^2} = 0 \quad (2.1)$$

kde  $\mu$  je permeabilita a  $\epsilon$  permitivita média v němž se vlna šíří. Pro harmonickou vlnu lze jako řešení psát:

$$\vec{E}(x, t) = A \cos[(kx - \omega t) - \beta_i] \vec{u} \quad (2.2)$$

kde  $A$  je amplituda vlny,  $k = \frac{2\pi}{\lambda}$  je vlnové číslo,  $\beta_i$  její počáteční fáze a  $\vec{u}$  jednotkový směrový vektor. Pokud nyní definujeme amplitudu vlny  $A$  jako komplexní, lze rovnici 2.2 vyjádřit v komplexním tvaru:

$$\vec{E}(x, t) = E \exp[i(kx - \omega t)] \vec{u} \quad (2.3)$$

kde  $E = A \exp(-i\beta_i)$  je komplexní amplituda vlny.

### 2.2 Vlna na rozhraní

Pro odvození podmínek spojitosti elektrické složky vlny dopadající na rozhraní je vhodné uvažovat integrační smyčku v blízkosti rozhraní (Obrázek 2.1). Vyjdeme z druhé Maxwellovy rovnice:

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \quad (2.4)$$

kde  $\vec{E}$  je vektor intenzity elektrického pole a  $\vec{B}$  vektor magnetické indukce. Integrováním přes plochu vyhraničenou smyčkou (Obrázek 2.1) a použitím Stokesovy věty dostáváme:

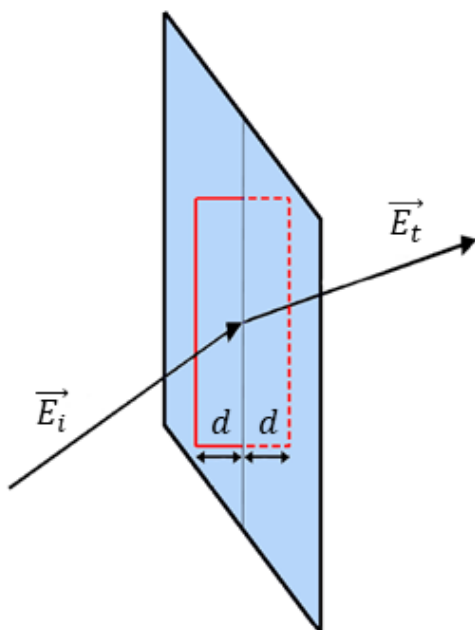
$$\int_S \nabla \times \vec{E} \cdot \vec{dS} = \oint_C \vec{E} \cdot \vec{dl} = -\frac{\partial}{\partial t} \int_S \vec{B} \cdot \vec{dS} \quad (2.5)$$

Nyní uvažujme případ, kdy se šířka smyčky  $C$  limitně blíží nule. V tomto případě se integrál magnetického pole blíží nule a rovnice 2.5 dostává tvar:

$$E_{i(t)}l - E_{t(t)}l = 0 \quad (2.6)$$

Tangenciální složka elektrického pole je na rozhraní spojitá:

$$E_{i(t)} = E_{t(t)} \quad (2.7)$$



Obrázek 2.1 Elektrické pole na rozhraní

Analogicky uvažujme smyčku v blízkosti rozhraní pro magnetické pole (Obrázek 2.2). Vyjdeme z první Maxwellovy rovnice:

$$\nabla \times \vec{H} = \vec{j} + \frac{\partial \vec{D}}{\partial t} \quad (2.8)$$

kde  $\vec{H}$  je vektor intenzity magnetického pole,  $\vec{D}$  vektor elektrické indukce a  $\vec{j}$  vektor proudové hustoty. Integrováním přes plochu vyhraničenou smyčkou (Obrázek 2.2) a použitím Stokesovy věty dostáváme:

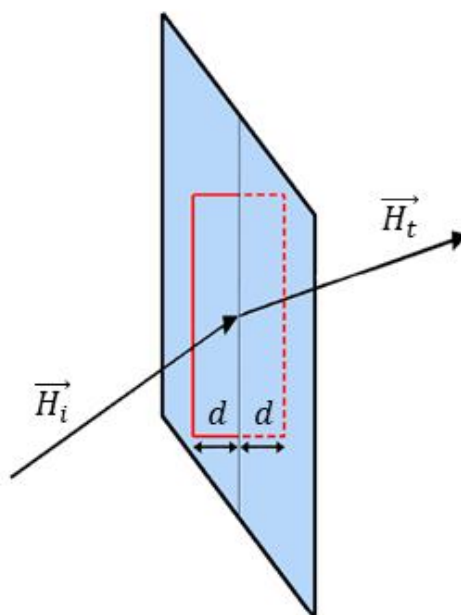
$$\int_S \nabla \times \vec{H} \cdot \vec{dS} = \oint_C \vec{H} \cdot \vec{dl} = \int_S \vec{j} \cdot \vec{dS} + \frac{\partial}{\partial t} \int_S \vec{D} \cdot \vec{dS} \quad (2.9)$$

Pokud opět uvažujeme šířku smyčky  $C$  limitně blíží se k nule, přechází rovnice 2.9 na tvar:

$$H_{i(t)}l - H_{t(t)}l = J_{S\perp}l \quad (2.10)$$

kde  $J_{S\perp}$  představuje povrchovou proudovou hustotu kolmou k tangenciální složce  $\vec{H}$ . Povrchová proudová hustota reprezentuje v tomto případě konečný proud v nekonečně malé vrstvě materiálu. Pokud tento materiál má konečnou vodivost, tak vrstva musí mít nekonečný odpor a protékající proud je nulový. Vztah 2.10 poté přechází na:

$$H_{i(t)} = H_{t(t)} \quad (2.11)$$



Obrázek 2.2 Magnetické pole na rozhraní

## 2.3 Fresnelovy rovnice

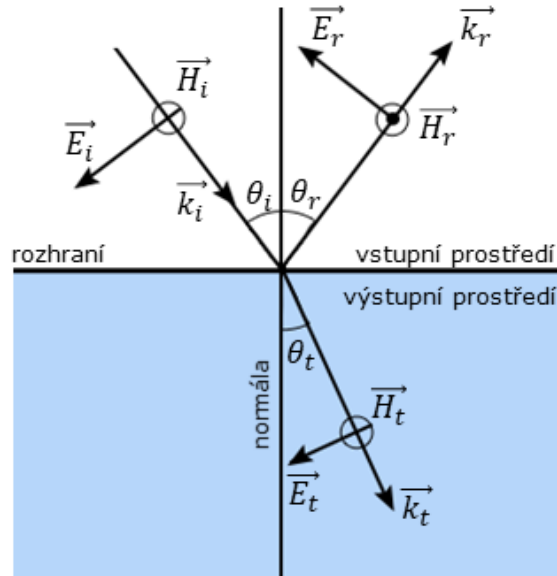
Fresnelovy rovnice popisují vztah mezi amplitudou, fází a polarizací vln vzniklých při interakci světla s rozhraním dvou prostupných materiálů s různými indexy lomu. Byly odvozeny Augustinem-Jeanem Fresnelem v roce 1823 a tvoří fundamentální základ klasické optiky.

Pro jejich odvození nejprve uvažujme rozhraní dvou materiálů (Obrázek 2.3) s dopadající vlnou popsanou vlnovým vektorem  $\vec{k}_i$  dopadajícím pod úhlem  $\theta_i$  k normále rozhraní. Tato vlna má amplitudu elektrického pole  $E_i$ .

Dle Snellova zákona daná vlna projde do druhého prostředí pod úhlem  $\theta_t$ , který lze získat ze vztahu:

$$\frac{\sin\theta_i}{\sin\theta_t} = \frac{n_2}{n_1} \quad (2.12)$$

kde  $n_1$  a  $n_2$  jsou indexy lomu zdrojového a cílového materiálu v daném pořadí. Úhel reflexe  $\theta_r$  je poté podle zákona reflexe rovinně souměrný k úhlu dopadu dle normály. Výsledná prošlá vlna má analogicky amplitudu  $E_t$  a odražená vlna amplitudu  $E_r$ .



Obrázek 2.3 P-polarizovaná vlna na rozhraní

Pro určení těchto amplitud vycházíme z okrajových podmínek elektrického pole  $\vec{E}$  a magnetického pole  $\vec{H}$  na rozhraní. Elektromagnetické pole dopadající na rozhraní lze rozdělit na dvě polarizační složky. Na P-polarizovanou vlnu s elektrickým polem v rovině tvořené vektorem  $\vec{k}_i$  a normálou rozhraní a S-polarizovanou vlnu k této rovině kolmou.

Prošlá a odražená vlna pro P-polarizovanou vlnu (Obrázek 2.3) bude identicky polarizovaná. Směry vektorů  $\vec{H}$ ,  $\vec{E}$  a  $\vec{k}_i$  odpovídají pravidlu pravé ruky (Obrázek 2.3). Okrajová podmínka pro elektrické pole je vyjádřena vztahem:

$$E_i \cos \theta_i + E_r \cos \theta_r = E_t \cos \theta_t \quad (2.13)$$

Vektory magnetického pole jsou rovnoběžné a platí pro ně:

$$H_i - H_r = H_t \quad (2.14)$$

Pro vyřešení těchto rovnic je vhodné zavést vztah mezi amplitudami elektrického a magnetického pole. Z Maxwellových rovnic víme, že vztah mezi nimi je následovný:

$$H = \sqrt{\frac{\varepsilon}{\mu}} E \quad (2.15)$$

kde  $\varepsilon$  je permitivita a  $\mu$  permeabilita materiálu v kterém se vlna šíří. Index lomu tohoto materiálu lze vyjádřit pomocí rychlosti světla ve vakuu  $c$ :

$$n = c\sqrt{\varepsilon\mu} \quad (2.16)$$

Po dosazení tohoto vztahu do rovnice 2.15 dostáváme:

$$H_{i,r} = n_1 \frac{E_i}{\mu_1 c} \quad H_t = n_2 \frac{E_t}{\mu_2 c} \quad (2.17)$$

Rovnice 2.14 dostává poté tvar:

$$\frac{n_1(E_i - E_r)}{\mu_1} = \frac{n_2 E_t}{\mu_2} \quad (2.18)$$

Reflexní a transmisní koeficient můžeme obecně pro amplitudy intenzity elektrického pole  $E$  definovat:

$$r = \frac{E_r}{E_i} \quad (2.19)$$

$$t = \frac{E_t}{E_i} \quad (2.20)$$

Kombinací rovnic 2.13, 2.18, 2.19 a 2.20 dostáváme Fresnelovy rovnice pro P polarizovanou vlnu:

$$t_p = \frac{2 \frac{n_1}{\mu_1} \cos \theta_i}{\frac{n_1}{\mu_1} \cos \theta_t + \frac{n_2}{\mu_2} \cos \theta_i} \quad (2.21)$$

$$r_p = \frac{\frac{n_1}{\mu_1} \cos \theta_t - \frac{n_2}{\mu_2} \cos \theta_i}{\frac{n_1}{\mu_1} \cos \theta_t + \frac{n_2}{\mu_2} \cos \theta_i} \quad (2.22)$$

Pro S-polarizovanou vlnu (Obrázek 2.4) mají rovnice 2.13 a 2.14 tvar:

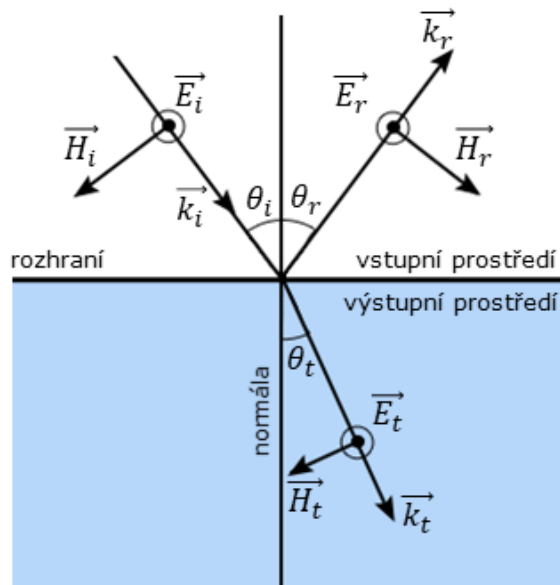
$$E_i + E_r = E_t \quad (2.23)$$

$$-H_i \cos \theta_i + H_r \cos \theta_i = -H_t \cos \theta_t \quad (2.24)$$

Výsledné Fresnelovy koeficienty pro S-polarizovanou vlnou jsou poté:

$$t_s = \frac{2 \frac{n_1}{\mu_1} \cos \theta_i}{\frac{n_1}{\mu_1} \cos \theta_i + \frac{n_2}{\mu_2} \cos \theta_t} \quad (2.25)$$

$$r_s = \frac{\frac{n_1}{\mu_1} \cos \theta_i - \frac{n_2}{\mu_2} \cos \theta_t}{\frac{n_1}{\mu_1} \cos \theta_i + \frac{n_2}{\mu_2} \cos \theta_t} \quad (2.26)$$



Obrázek 2.4 S-polarizovaná vlna na rozhraní

## 2.4 Intenzita reflexe a transmise

Z experimentálního hlediska je vhodnější vyjádření koeficientů v intenzitním tvaru. Pro vlnu šířící se v nemagnetickém médiu s indexem lomu  $n$  lze (časovou) střední hodnotu intenzity elektromagnetického záření definovat:

$$I = 2nc\varepsilon_0|E|^2 \quad (2.27)$$

kde  $E$  je její amplituda a  $\varepsilon_0$  permitivita vakua. Odražená vlna se šíří identickým médiem jako vlna dopadající a pro její intenzitní reflexní koeficient můžeme psát:

$$R = \frac{|E_r|^2}{|E_i|^2} = |r|^2 \quad (2.28)$$

Při výpočtu intenzitního transmisního koeficientu prošlé vlny musíme zohlednit skutečnost, že vlna prošlá se šíří jiným médiem s indexem lomu  $n_2$ . Transmisní koeficient lze navíc pro intenzitu vyjádřit pouze mezi dvěma neabsorbujícími prostředími. Intenzita je v tomto případě počítána na jednotku plochy vlnoplochy. Jelikož jsou jednotlivé vlnoplochy dopadající a prošlé vlny vzhledem k rozhraní pod úhly  $\theta_i$  a  $\theta_t$ . Intenzitní transmisní koeficient má tvar [2]:

$$T = \frac{n_2 \cos \theta_t |E_t|^2}{n_1 \cos \theta_i |E_i|^2} = \frac{n_2 \cos \theta_t}{n_1 \cos \theta_i} |t|^2 \quad (2.29)$$

## 2.5 Změna fáze na rozhraní

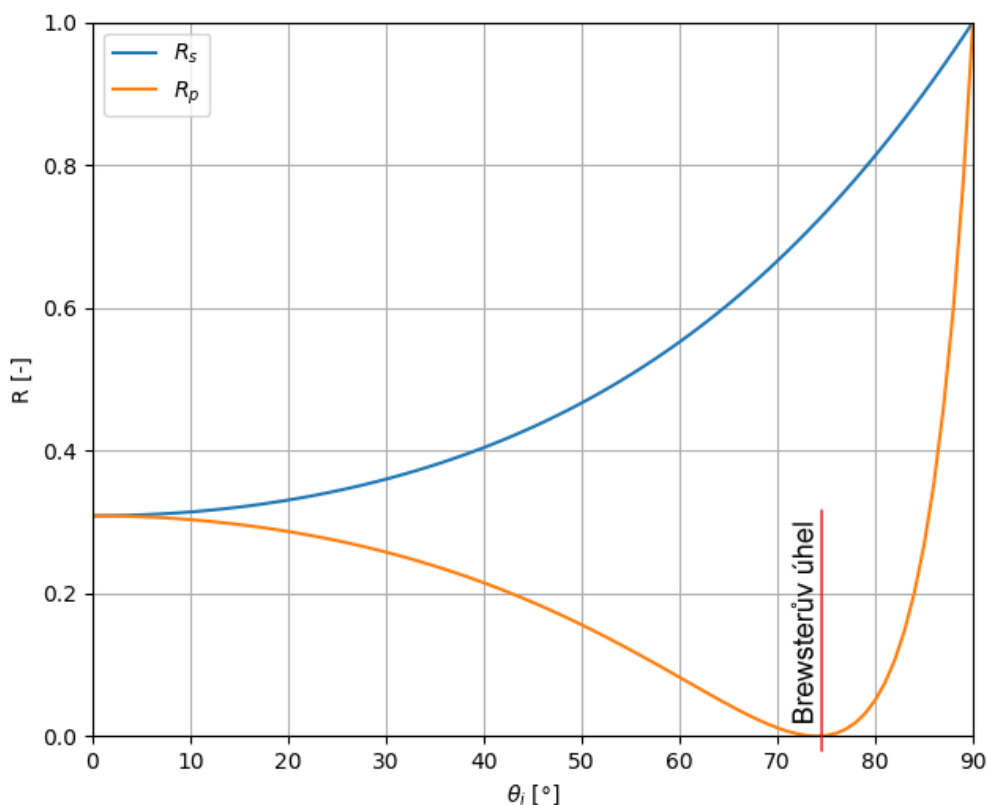
Pro rozhraní neabsorbujících prostředí lze odvodit zákonitosti změny fáze na rozhraní. Z rovnic 2.22 a 2.26 je patrné, že při úhlu dopadající vlny blízkém normále je hodnota reflexního koeficientu záporná když  $n_1 < n_2$ . To znamená, že fáze elektrického

pole vlny se posune o  $180^\circ$  při reflexi na rozhraní s prostředím s vyšším indexem lomu. Pro S-polarizovanou vlnu je znaménko reflexního koeficientu stejné pro všechny úhly dopadu. Pro P-polarizovanou vlnu se znaménko mění při překročení Brewsterova úhlu. Transmisní koeficienty (2.8, 2.12) jsou naopak vždy kladné a prošlá vlna nemá fázový posun vůči vlně dopadající. Pokud se jedná o absorbující prostředí, dochází vždy k fázovému posunu. [3]

## 2.6 Brewsterův úhel

Z níže uvedeného obrázku (Obrázek 2.5) je patrné, že intenzita odražené P-polarizované vlny na rozhraní vzduch křemík klesá k nule když  $\theta_t + \theta_i = 90^\circ$ . Intenzitní reflexní koeficient odražené P-polarizované vlny (2.22) má tvar:

$$R_p = |r_p|^2 = \frac{\tan^2(\theta_i - \theta_t)}{\tan^2(\theta_i + \theta_t)} \quad (2.30)$$



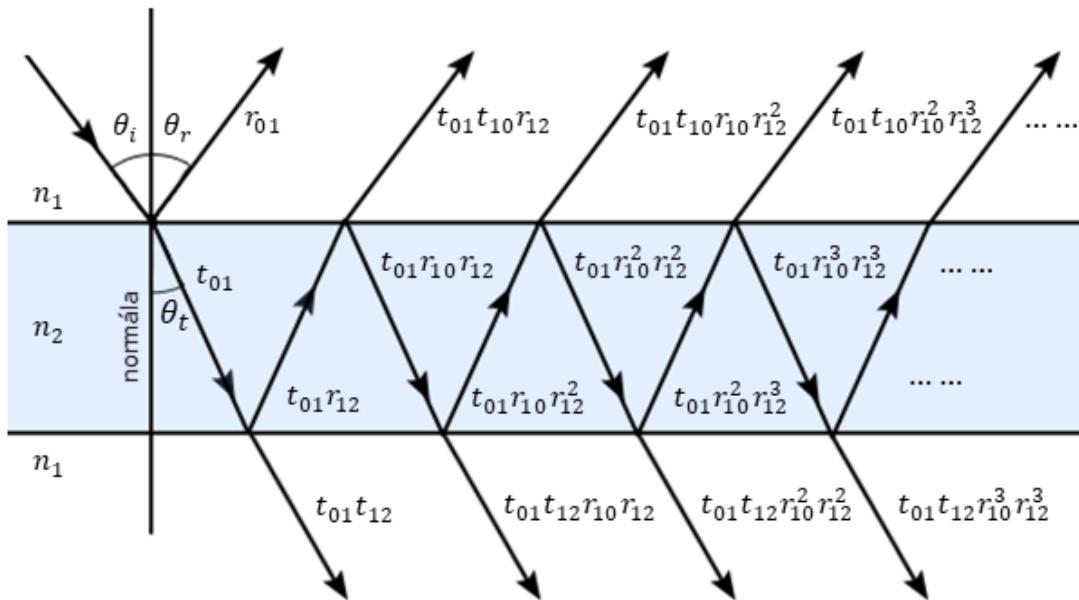
Obrázek 2.5 Brewsterův úhel na rozhraní vzduch-křemík

Při součtu úhlů dopadající a prošlé vlny rovném  $90^\circ$  se blíží jmenovatel rovnice 2.30 nekonečnu a intenzita reflexe výsledně limitně klesá k nule. Úhel dopadající vlny, při kterém k tomuto dochází, je nazýván Brewsterův úhel  $\theta_B$  a ze Snellova zákona (2.12) lze pro něj psát [4]:

$$\tan \theta_B = \frac{n_2}{n_1} \quad (2.31)$$

## 2.7 Jednovrstvá struktura

K ilustraci významu maticového výpočtu je vhodné nejprve odvodit vztahy pro neabsorbující vrstvu umístěnou v neabsorbujícím prostředí. V této situaci dopadá vstupní vlna o vlnové délce  $\lambda_i$  z neabsorbujícího prostředí s indexem lomu  $n_1$  na rozhraní prostředí-vrstva pod úhlem  $\theta_i$ . Zde dochází k částečnému odrazu pod úhlem  $\theta_r$  a průchodu vlny pod úhlem  $\theta_t$ . Prošlá vlna následně putuje vrstvou s tloušťkou  $d$ , indexem lomu  $n_2$  a dopadá na druhé rozhraní vrstva-prostředí. Zde opět dochází k částečnému odrazu a průchodu. Z níže uvedeného obrázku (Obrázek 2.6) je patrné, že proces vnitřních odrazů ve vrstvě je nekonečný.



Obrázek 2.6 Jednovrstvá struktura

Pro odvození vztahů využijeme Fresnelových koeficientů odvozených v kapitole 2.3. Reflexní a transmisní Fresnelův koeficient pro rozhraní prostředí-vrstva označíme  $r_{01}$ ,  $t_{01}$  pro směr dopadající vlny a  $r_{10}$ ,  $t_{10}$  pro směr obrácený. Obdobně pro rozhraní vrstva-prostředí  $r_{12}$ ,  $t_{12}$  a  $r_{21}$ ,  $t_{21}$ . V případě několikanásobných reflexí je rozdíl optické dráhy sousedících vln  $\Delta$  definován:

$$\Delta = 2n_1d \cos \theta_t \quad (2.32)$$

Vlivem polarizace materiálu se vlna vrstvou pohybuje pomaleji nežli prostředím pokud platí, že  $n_2 > n_1$ . Dochází tedy k fázovému posunu vlny  $\beta_m$ , kde  $m$  je celkové množství předešlých vnitřních odrazů:



$$\beta_m = (m + 1) \frac{2\pi\Delta}{\lambda} \quad (2.33)$$

Výsledná amplituda odražené vlny je definována součtem nekonečné série:

$$\begin{aligned} \frac{E_r}{E_i} &= r_1 + t_{01}r_{12}t_{10}e^{-2i\beta} \sum_{n=1}^{\infty} (-1)^n (r_{12}r_{10})^n e^{-2in\beta} \\ &= r_1 + \frac{t_{01}r_{12}t_{10}e^{-2i\beta}}{1 + r_{01}r_{12}e^{-2i\beta}} = \frac{r_{01} + r_{12}e^{-2i\beta}}{1 + r_{01}r_{12}e^{-2i\beta}} \end{aligned} \quad (2.34)$$

Pro amplitudu prošlé vlny obdobným postupem dostaneme:

$$\frac{E_t}{E_i} = \frac{t_{01}t_{12}e^{-i\beta}}{1 + r_{01}r_{12}e^{-2i\beta}} \quad (2.35)$$

Použitím vzorců pro intenzitu (2.28 a 2.29) dostáváme finální vztahy pro celkovou intenzitu odražené a prošlé vlny:

$$R = \frac{r_{01}^2 + r_{12}^2 + 2r_{01}r_{12} \cos 2\beta}{1 + r_{01}^2 r_{12}^2 + 2r_{01}r_{12} \cos 2\beta} \quad (2.36)$$

$$T = \frac{n_2}{n_1} \frac{t_{01}^2 t_{12}^2}{1 + r_{01}^2 r_{12}^2 + 2r_{01}r_{12} \cos 2\beta} \quad (2.37)$$

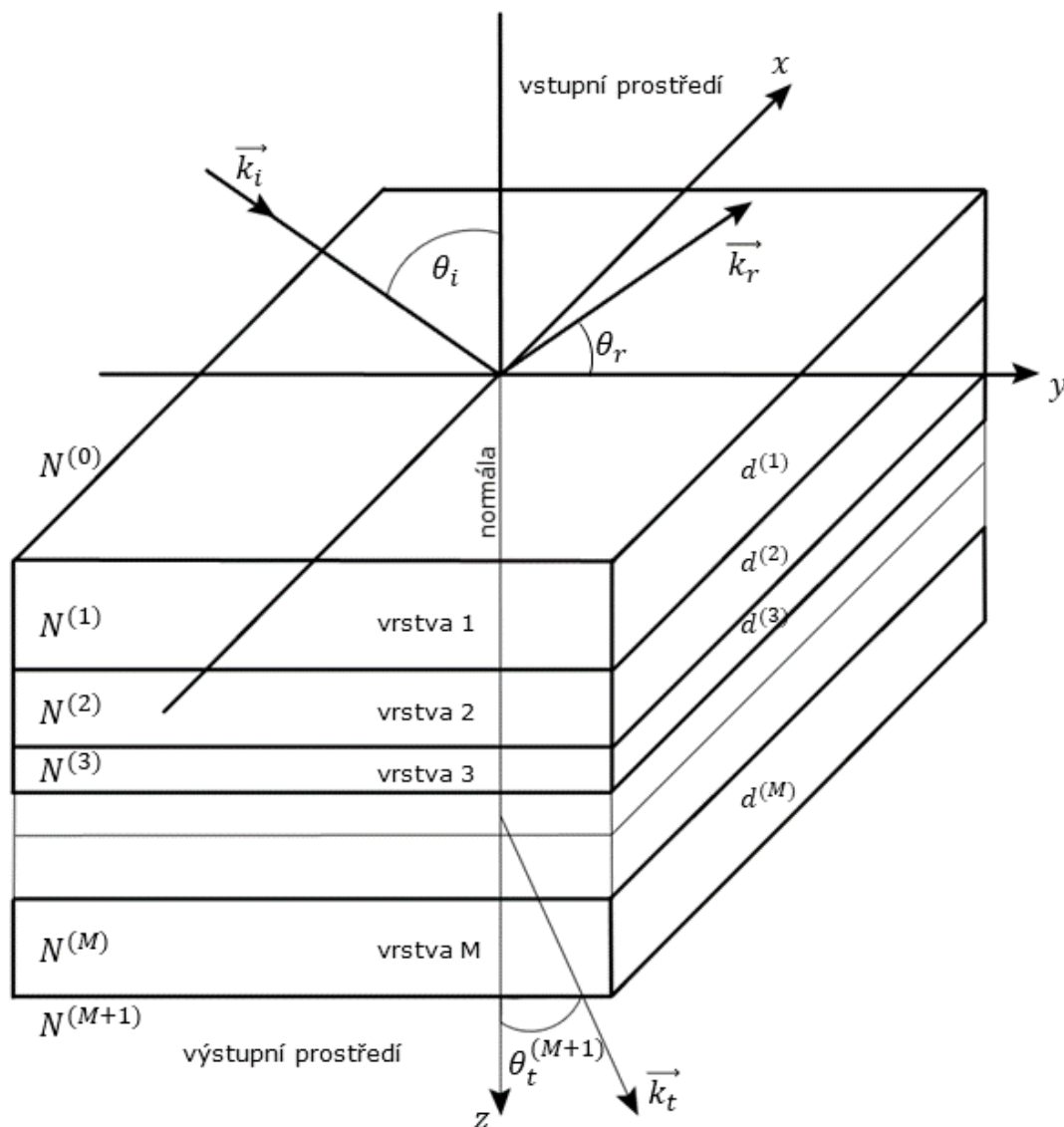
## 2.8 Mnohovrstvá struktura

Uvažujme nyní strukturu složenou z více vrstev značených sekvenčně 1, 2, ...,  $m$ , ...,  $M$ . Tyto vrstvy jsou k sobě vzájemně paralelní a umístěny mezi prostředí tvořící poloprostor vstupní (0) a výstupní ( $M + 1$ ). Všechna prostředí dále považujeme za izotropní a homogenní. Vrstvy jsou charakterizovány svojí tloušťkou  $d^{(m)}$  a indexem lomu  $N^{(m)}$ . Jelikož vrstvy mohou být absorbujičho charakteru, zavádíme pro výpočet komplexní index lomu, kde je útlum v materiálu reprezentován extinkčním koeficientem  $\kappa$ :

$$N = n - i\kappa \quad (2.38)$$

Na rozhraní dopadá z prostředí (0) monochromatická vlna pod úhlem  $\theta_i$  s polarizací typu S nebo P. V témže prostředí je vytvořena vlna odražená pod úhlem  $\theta_r$  a v prostředí ( $M + 1$ ) vlna lomená pod úhlem  $\theta_t$ . Pole uvnitř  $i$ -té vrstvy vybuzené dopadající vlnou lze rozdělit na dvě složky: vlna šířící se souhlasně s dopadající vlnou a vlna šířící se ve směru opačném. Pro přehlednost výpočtu je souhlasná vlna značena indexem + a protichůdná indexem -.

Rovinné vlny vytvořené dopadající vlnou v libovolném místě struktury zachovávají polarizaci zdrojové vlny. V níže uvedených výpočtech vždy předpokládáme předem definovanou polarizaci počítané vlny.



Obrázek 2.7 Izotropní mnohovrstvá struktura

Celkové pole  $\mathbf{E}$  v libovolné rovině  $z$  lze popsat sloupcovým vektorem za pomoci komplexních amplitud elektrického pole vlny souhlasné  $\varepsilon^+$  a protichůdné  $\varepsilon^-$ :

$$\mathbf{E}(z) = \begin{pmatrix} \varepsilon^+(z) \\ \varepsilon^-(z) \end{pmatrix} \quad (2.39)$$

Nyní uvažujme dvojici rovnoběžných rozhraní  $z'$  a  $z''$ . Jelikož výsledná vlna je součtem příspěvků úměrných dopadajícím vlnám, musí existovat lineární transformační matice  $\mathbf{S}$  rozměru  $2 \times 2$  svazující  $\mathbf{E}(z'')$  a  $\mathbf{E}(z')$ :

$$\mathbf{E}(z') = \mathbf{S}\mathbf{E}(z'') \quad (2.40)$$

$$\begin{pmatrix} \varepsilon^+(z') \\ \varepsilon^-(z') \end{pmatrix} = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix} \begin{pmatrix} \varepsilon^+(z') \\ \varepsilon^-(z') \end{pmatrix} \quad (2.41)$$

Matice  $\mathbf{S}$  vždy charakterizuje strukturu mezi dvěma rovnoběžnými rovinami. Pokud si tyto roviny definujeme jako opačné strany rozhraní mezi vrstvami  $m - 1$  a  $m$ , můžeme pro rovinu nekonečně blízkou rozhraní  $z^{(m-1)}$  psát:

$$\lim_{z \rightarrow z_-^{(m-1)}} \mathbf{E}(z) = \mathbf{I}^{(m-1,m)} \lim_{z \rightarrow z_+^{(m-1)}} \mathbf{E}(z), \quad (2.42)$$

kde  $\mathbf{I}^{(m-1,m)}$  je matice o rozměru  $2 \times 2$  charakterizující rozhraní mezi vrstvou  $m$  a  $m - 1$ . Stejným principem můžeme tyto roviny definovat uvnitř vrstvy, kterou vlna prochází. V tomto případě matice  $\mathbf{L}^{(m)}$  charakterizuje danou vrstvu  $m$ :

$$\lim_{z \rightarrow z_+^{(m-1)}} \mathbf{E}(z) = \mathbf{L}^{(m)} \lim_{z \rightarrow z_+^{(m-1)}} \mathbf{E}(z), \quad (2.43)$$

Pro simulace je vhodné definovat vztah celkové odražené a prošlé vlny vůči vlně dopadající. Pokud si zvolíme naše referenční roviny v těsné blízkosti vstupního prostředí (0) a výstupního ( $M + 1$ ) dostáváme rovnici:

$$\lim_{z \rightarrow z_-^{(0)}} \mathbf{E}(z) = \mathbf{S} \lim_{z \rightarrow z_+^{(M)}} \mathbf{E}(z) \quad (2.44)$$

Matice  $\mathbf{S}$  představuje rozptylovou matici charakterizující odraženou a prošlou vlnu mnohovrstvou strukturou. Výpočetně jí lze vyjádřit jako součin jednotlivých matic rozhraní  $\mathbf{I}^{(m-1,m)}$  a matic vrstev  $\mathbf{L}^{(m)}$ :

$$\mathbf{S} = \mathbf{I}^{(01)} \mathbf{L}^{(1)} \mathbf{I}^{(12)} \mathbf{L}^{(2)} \dots \mathbf{I}^{(m-1,m)} \mathbf{L}^{(m)} \dots \mathbf{L}^{(M)} \mathbf{I}^{(M,M+1)} \quad (2.45)$$

K určení rozptylové matice je tedy nutné nejprve definovat matice jednotlivých rozhraní a vrstev. Pro matici rozhraní  $\mathbf{I}^{(ab)}$  mezi libovolnými vrstvami  $a$  a  $b$  lze obecně psát [5]:

$$\begin{pmatrix} \varepsilon_a^+ \\ \varepsilon_a^- \end{pmatrix} = \begin{pmatrix} I_{11}^{(ab)} & I_{12}^{(ab)} \\ I_{21}^{(ab)} & I_{22}^{(ab)} \end{pmatrix} \begin{pmatrix} \varepsilon_b^+ \\ \varepsilon_b^- \end{pmatrix} \quad (2.46)$$

Pro určení prvků této matice uvažujme nejprve případ, kdy na rozhraní  $ab$  dopadá vlna  $\varepsilon_a^+$  z prostředí  $a$ . Komplexní amplituda vlny odražené  $\varepsilon_a^-$  z tohoto rozhraní a vlny prošlé  $\varepsilon_b^+$  lze za pomoci Fresnelových koeficientů  $t^{(ab)}$  a  $r^{(ab)}$  definovat následovně:

$$\varepsilon_b^+ = t^{(ab)} \varepsilon_a^+ \quad (2.47)$$

$$\varepsilon_a^- = r^{(ab)} \varepsilon_a^+ \quad (2.48)$$

Jelikož na rozhraní dopadá vlna pouze v jednom směru, můžeme člen rovnice 2.46  $\varepsilon_b^-$ , představující světlo vstupující na rozhraní z opačného směru, položit roven nule. Poté píšeme:

$$\begin{pmatrix} \varepsilon_a^+ \\ \varepsilon_a^- \end{pmatrix} = \begin{pmatrix} I_{11}^{(ab)} & I_{12}^{(ab)} \\ I_{21}^{(ab)} & I_{22}^{(ab)} \end{pmatrix} \begin{pmatrix} \varepsilon_b^+ \\ 0 \end{pmatrix} \quad (2.49)$$

a pro vstupní sloupcový vektor můžeme psát:

$$\varepsilon_a^+ = I_{11}^{(ab)} \varepsilon_b^+ \quad (2.50)$$

$$\varepsilon_a^- = I_{21}^{(ab)} \varepsilon_b^+ \quad (2.51)$$

Srovnáním těchto rovnic s rovnicemi pro amplitudy prošlé (2.47) a odražené (2.48) vlny dostáváme pro dva z prvků matice rozhraní  $I^{(ab)}$ :

$$I_{11}^{(ab)} = \frac{1}{t^{(ab)}} \quad (2.52)$$

$$I_{21}^{(ab)} = \frac{r^{(ab)}}{t^{(ab)}} \quad (2.53)$$

V druhém případě dopadá na rozhraní  $ab$  pouze vlna  $\varepsilon_b^-$  z prostředí  $b$ . Opět vyjádříme komplexní amplitudu vlny odražené  $\varepsilon_b^+$  a vlny prošlé  $\varepsilon_a^-$  pomocí Fresnelových koeficientů  $r^{(ba)}$  a  $t^{(ba)}$ :

$$\varepsilon_b^+ = r^{(ba)} \varepsilon_b^- \quad (2.54)$$

$$\varepsilon_a^- = t^{(ba)} \varepsilon_b^- \quad (2.55)$$

V tomto případě dopadá vlna pouze ve směru  $-$ . Člen rovnice 2.46  $\varepsilon_a^+$  můžeme položit roven nule. Rovnice má poté tvar:

$$\begin{pmatrix} 0 \\ \varepsilon_a^- \end{pmatrix} = \begin{pmatrix} I_{11}^{(ab)} & I_{12}^{(ab)} \\ I_{21}^{(ab)} & I_{22}^{(ab)} \end{pmatrix} \begin{pmatrix} \varepsilon_b^+ \\ \varepsilon_b^- \end{pmatrix} \quad (2.56)$$

dosazením pro prvky sloupcových vektorů dostáváme:

$$\varepsilon_b^+ = -I_{12}^{(ab)} t^{(ab)} \varepsilon_b^- \quad (2.57)$$

$$\varepsilon_a^- = \frac{r^{(ab)}}{t^{(ab)}} \varepsilon_b^+ + I_{22}^{(ab)} \varepsilon_b^- \quad (2.58)$$

Pokud následně dosadíme do těchto rovnic vztahy pro komplexní amplitudy odražené (2.54) a prošlé (2.55) vlny dostáváme:

$$I_{12}^{(ab)} = -\frac{r^{(ba)}}{t^{(ab)}} \quad (2.59)$$

$$I_{22}^{(ab)} = t^{(ba)} - \frac{r^{(ab)}r^{(ba)}}{t^{(ab)}} \quad (2.60)$$

Rovnici 2.60 lze značně zjednodušit za pomoci vzájemných vztahů mezi Fresnelovými koeficienty pro opačné směry šíření:

$$r^{(ab)} = -r^{(ba)} \quad (2.61)$$

$$t^{(ab)}t^{(ba)} - r^{(ab)}r^{(ba)} = 1 \quad (2.62)$$

Kombinací předchozích rovnic dostáváme definici matice rozhraní  $\mathbf{I}^{(ab)}$  pomocí Fresnelových reflexních a transmisních koeficientů:

$$\mathbf{I}^{(ab)} = \begin{pmatrix} 1 & r^{(ab)} \\ \frac{1}{t^{(ab)}} & \frac{r^{(ab)}}{t^{(ab)}} \\ r^{(ab)} & 1 \\ \frac{r^{(ab)}}{t^{(ab)}} & \frac{1}{t^{(ab)}} \end{pmatrix} \quad (2.63)$$

Druhou složkou rozptylové matice je matice vrstvy  $\mathbf{L}^{(m)}$ . V libovolné vrstvě o komplexním indexu lomu  $N^{(m)}$  a tloušťce  $d^{(m)}$  lze v každém místě definovat dvě vlny šířící se protichůdně ve směrech  $+$  a  $-$ . Výsledné pole těchto vln v rovině  $z$  lze popsat pomocí rovnice rovinné vlny (2.3):

$$\varepsilon^\pm(z) = \varepsilon^\pm(0)e^{i\omega t} \exp\left[-i\frac{2\pi}{\lambda}yN(z)\sin\theta(z)\right] \exp\left[\mp i\frac{2\pi}{\lambda}zN(z)\cos\theta(z)\right] \quad (2.64)$$

kde  $\lambda$  je vlnová délka vlny o úhlové frekvenci  $\omega$  šířící se ve vakuu. Dle goniometrických funkcí lze tuto rovnici (2.51) rozdělit na podélnou složku  $\frac{2\pi}{\lambda}N(z)\sin\theta(z)$  a normálovou složku  $\pm\frac{2\pi}{\lambda}N(z)\cos\theta(z)$ . Jelikož je dle Snellova zákona součin  $N(z)\sin\theta(z)$  neměnný v celé struktuře, můžeme pro další výpočet uvažovat pouze normálovou složku.

Pokud nyní definujeme novou rovinu uvnitř stejné vrstvy  $z^{(m-1)}$  vzdálenou od roviny  $z^{(m)}$  tak, že  $z^{(m)} = z^{(m-1)} + d^{(m)}$ , můžeme pro výsledné vlny v těchto rovinách psát:

$$\varepsilon^\pm(z^{(m)}) = \varepsilon^\pm(0)\exp\left[\mp i\frac{2\pi}{\lambda}N^{(m)}(z^{(m-1)} + d^{(m)})\cos\theta^{(m)}\right] \quad (2.65)$$

$$\varepsilon^\pm(z^{(m-1)}) = \varepsilon^\pm(0)\exp\left[\mp i\frac{2\pi}{\lambda}N^{(m)}z^{(m-1)}\cos\theta^{(m)}\right] \quad (2.66)$$

Kde  $\cos\theta^{(m)}$  vyjadřuje úhel šíření vlny danou vrstvou ve vztahu k normále. Kombinací rovnic 2.52 a 2.53 můžeme vyjádřit vztah:

$$\varepsilon^\pm(z^{(m)}) = \varepsilon^\pm(z^{(m-1)}) \exp \left[ \mp i \frac{2\pi}{\lambda} N^{(m)} d^{(m)} \cos \theta^{(m)} \right] \quad (2.67)$$

Pokud tuto rovnici vyjádříme pro jednotlivé směry a převedeme do maticové podoby, dostaneme:

$$\begin{pmatrix} \varepsilon^+(z^{(m-1)}) \\ \varepsilon^-(z^{(m-1)}) \end{pmatrix} = \begin{pmatrix} e^{i\beta^{(m)}} & 0 \\ 0 & e^{-i\beta^{(m)}} \end{pmatrix} \begin{pmatrix} \varepsilon^+(z^{(m)}) \\ \varepsilon^-(z^{(m)}) \end{pmatrix} \quad (2.68)$$

kde  $\beta^{(m)}$  vyjadřuje bezrozměrný parametr:

$$\beta^{(m)} = \frac{2\pi}{\lambda} N^{(m)} d^{(m)} \cos \theta^{(m)} \quad (2.69)$$

Matice šíření ve vrstvě  $\mathbf{L}^{(m)}$  má poté tvar:

$$\mathbf{L}^{(m)} = \begin{pmatrix} e^{i\beta^{(m)}} & 0 \\ 0 & e^{-i\beta^{(m)}} \end{pmatrix} \quad (2.70)$$

Dosazením výsledných vztahů pro matice rozhraní  $\mathbf{I}^{(ab)}$  (2.63) a vrstev  $\mathbf{L}^{(m)}$  (2.70) do rovnice 2.45 můžeme na výslednou rozptylovou matici struktury pohlížet jako na matici jednoho rozhraní:

$$\begin{pmatrix} \varepsilon^+(z^{(0)}) \\ \varepsilon^-(z^{(0)}) \end{pmatrix} = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix} \begin{pmatrix} \varepsilon^+(z^{(M)}) \\ \varepsilon^-(z^{(M)}) \end{pmatrix} \quad (2.71)$$

Pokud opět uvážíme podmínku, kdy na strukturu dopadá vlna pouze v souhlasném směru (+) z prostředí (0), můžeme člen  $\varepsilon^-(z^{(M)})$  položit roven nule. Výsledné zjednodušené maticové rovnice mají tvar:

$$\varepsilon^+(z^{(0)}) = S_{11} \varepsilon^+(z^{(M)}) \quad (2.72)$$

$$\varepsilon^-(z^{(0)}) = S_{21} \varepsilon^+(z^{(M)}) \quad (2.73)$$

Pokud prvky matice  $\mathbf{S}$  představují rozptylovou matici celkové struktury, lze pro danou strukturu definovat celkový reflexní koeficient  $r^{(0,M+1)}$  a celkový transmisní koeficient  $t^{(0,M+1)}$  pomocí maticových rovnic (2.72, 2.73):

$$r^{(0,M+1)} = \frac{\varepsilon^-(z^{(0)})}{\varepsilon^+(z^{(0)})} = \frac{S_{21}}{S_{11}} \quad (2.74)$$

$$t^{(0,M+1)} = \frac{\varepsilon^+(z^{(M)})}{\varepsilon^+(z^{(0)})} = \frac{S_{21}}{S_{11}} \quad (2.75)$$

Nyní uvažujme strukturu s jednou vrstvou (1) mezi prostředními (0) a (2). Rozptylová matice je dle rovnice (2.45) definována:

$$\mathbf{S} = \mathbf{I}^{(01)}\mathbf{L}^{(1)}\mathbf{I}^{(12)} \quad (2.76)$$

Pro matice rozhraní  $\mathbf{I}^{(01)}$ ,  $\mathbf{I}^{(12)}$  dosadíme z rovnice (2.63) a pro matici vrstvy  $\mathbf{L}^{(1)}$  z rovnice (2.70):

$$\mathbf{S} = \begin{pmatrix} 1 & r^{(01)} \\ \frac{t^{(01)}}{r^{(01)}} & \frac{1}{t^{(01)}} \end{pmatrix} \begin{pmatrix} e^{i\beta^{(1)}} & 0 \\ 0 & e^{-i\beta^{(1)}} \end{pmatrix} \begin{pmatrix} 1 & r^{(12)} \\ \frac{t^{(12)}}{r^{(12)}} & \frac{1}{t^{(12)}} \end{pmatrix} \quad (2.77)$$

roznásobením a vytknutím:

$$\mathbf{S} = \frac{e^{i\beta^{(1)}}}{t^{(01)}t^{(12)}} \begin{pmatrix} 1 + r^{(01)}r^{(12)}e^{-2i\beta^{(1)}} & r^{(12)} + r^{(01)}e^{-2i\beta^{(1)}} \\ r^{(01)} + r^{(12)}e^{-2i\beta^{(1)}} & r^{(01)}r^{(12)} + e^{-2i\beta^{(1)}} \end{pmatrix} \quad (2.78)$$

Při zachování podmínky dopadu vlny na rozhraní (01) pouze z jednoho směru stačí vyjádřit z této matice prvky  $S_{11}$  a  $S_{21}$ :

$$S_{11} = \frac{e^{i\beta^{(1)}}}{t^{(01)}t^{(12)}} (1 + r^{(01)}r^{(12)}e^{-2i\beta^{(1)}}) \quad (2.79)$$

$$S_{21} = \frac{e^{i\beta^{(1)}}}{t^{(01)}t^{(12)}} (r^{(01)} + r^{(12)}e^{-2i\beta^{(1)}}) \quad (2.80)$$

po dosazení do rovnic pro celkové koeficienty (2.74) a (2.75):

$$r^{(0,2)} = \frac{r^{(01)} + r^{(12)}e^{-2i\beta^{(1)}}}{1 + r^{(01)}r^{(12)}e^{-2i\beta^{(1)}}} \quad (2.81)$$

$$t^{(0,2)} = \frac{t^{(01)}t^{(12)}e^{-i\beta^{(1)}}}{1 + r^{(01)}r^{(12)}e^{-2i\beta^{(1)}}} \quad (2.82)$$

Tyto rovnice jsou identické k těm odvozeným pro sumaci vícenásobně odražených vln (2.34) a (2.35) v předchozí kapitole.





## 3 Programovací jazyk Python

Python je interpretovaný objektově orientovaný jazyk vyvíjený pod open-source licencí. V současnosti je po boku Javy, C# a Javascriptu nejpoužívanější programovací jazyk a pokud se omezíme na pole matematických simulací, má konkurenci pouze v jazyku R. Právě díky rozvoji polí datové analýzy, matematických simulací a strojového učení je Python jedním z nejrychleji rostoucích jazyků posledního desetiletí.

Pro práci byl zvolen především pro jeho dostupnost a rozšiřitelnost. Množství externích open-source knihoven zajišťuje možnost propojení programu s libovolnou externí aplikací a zároveň jednoduchost syntaxe Pythonu umožňuje případné přenesení kódu do jiného vhodného prostředí (Mathematica, MATLAB).

### 3.1 Historie

Historie Pythonu je úzce spjatá s programovacím jazykem ABC, vyvíjeným v institutu CWI (Centrum Wiskunde & Informatica). Zde kolem roku 1980 zaměstnanec Guido van Rossum vyvinul v reakci na každodenní frustraci s prací v jazyce ABC první náčrt jazyku Python. Tento jazyk měl dle jeho slov zachovat fungující prvky jazyku ABC, doplnit funkcionalitu a zjednodušit strukturu syntaxe.

Začátek implementace se datuje k roku 1989 a v únoru 1992 byla vydána první veřejná verze 0.9.0. Jméno jazyka Python vzešlo z názvu britské komediální skupiny Monthy Python, které byl Rossum velkým příznivcem.



*Obrázek 3.1 Logo jazyka Python [6]*

Ve verzi 1.0 byl Python vydán v lednu roku 1994 a v roce 2000 se jeho vývoj odtrhnul od společnosti CW pod BeOpen PythonLabs, kde započala jeho dráha jako open-source, tedy volně dostupného jazyka. Spolu s tímto krokem byla představena verze 2.0. S přechodem do verze 3.0 roku 2008 byla přerušena zpětná kompatibilita mezi verzemi, tedy kód psaný v Pythonu verze 2.0 nelze bez modifikací spustit v Pythonu verze 3.0.

K datu psaní práce je Python aktuálně distribuován ve verzi 3.6 vydané v prosinci roku 2016.

## 3.2 Funkcionalita

Python je multiparadigmatický programovací jazyk s plnou možností logického a objektově orientovaného programování. Řada jeho funkcí také podporuje funkcionální či aspektově-orientované programování.

### 3.2.1 Datové typy a proměnné

Python využívá principu Duck-typing popsaného větou Jamese Whitcomba Ryileého „*Pokud to chodí jako kachna a kváká jako kachna, tak to musí být kachna*“. V realitě to znamená, že každý objekt v Pythonu může být používán v libovolném kontextu do momentu kdy je použit stylem, který sám nepodporuje.

```
class Papousek:
    def letat(self):
        print("Papousek leti")

class Meteor:
    def letat(self):
        print("Meteor leti")

class Zelva:
    def plavat(self):
        print("Zelva plave")

def vzlet(entity):
    entity.letat()

papousek = Papousek()
meteor = Meteor()
zelva = Zelva()

vzlet(papousek) # vytiskne "Papousek leti"
vzlet(meteor) # vytiskne "Meteor leti"
vzlet(zelva) # zobrazí error `Zelva' objekt nema metodu 'letat`
```

#### *Ukázka kódu 3.1 Duck-typing v jazyce Python*

Z toho vyplývá důležitá vlastnost Pythonu, jeho dynamické typování. Proměnné nejsou přímo typově definovány při jejich vytvoření a samotná kontrola probíhá až při kompilaci. I přes dynamičnost je ale Python silně typován. Pokud dojde k zavolání nedefinované operace, například sečtení textu a čísla, je rovnou zobrazen chybový stav namísto pokusu danou operaci vykonat.

Pro potřeby práce jsou důležité především datové typy číselné (Tabulka 3.1). Pro popis množiny celých čísel je v Pythonu zabudován datový typ `int`. Oproti většině programovacích jazyků není v případě Pythonu verze 3 a výše tato množina dělena na malá a velká čísla. Jeho maximální velikost je dána limity interpretu.

*Tabulka 3.1 Základní číselné typy*

Typ	Popis	Příklad deklarace
<code>int</code>	celé číslo	<code>x = 3</code>
<code>float</code>	racionální číslo	<code>x = 2,0</code>
<code>complex</code>	komplexní číslo	<code>x = 1,5 + 5j</code>

Množina racionálních čísel je reprezentována datovým typem `float`. Zde je nutné zmínit nedostatek počítačové reprezentace této množiny. Pro příklad desetinné číslo 0,125 má hodnotu:

$$0,125 = \frac{1}{10} + \frac{2}{100} + \frac{5}{1000}$$

To samé číslo v binární soustavě má hodnotu 0,001 a pokud ho analogicky zapíšeme jako součet binárních zlomků:

$$0,001 = \frac{0}{2} + \frac{0}{4} + \frac{1}{8}$$

Při běhu programu musí být každé desetinné číslo uloženo jako binární. Při této reprezentaci často dojde k situaci, kdy desetinné číslo v desítkové soustavě je možné popsat pouze periodicky se opakující sekvencí v binární soustavě. Desítkové desetinné číslo 0,1 je v binární soustavě následovně:

$$(0,1)_{10} = (0,00011001100110011001100110011001100110011001100110011001100110011...)_{2}$$

Tato skutečnost vnáší do výpočtů s desetinnými čísly vnořenou chybu, jelikož řada desetinných čísel je pouze jejich velice blízkou aproximací.

Pro práci s komplexními čísly slouží modul `cmath` využívající mimo standardních číselných typů komplexní datový typ `complex`. Vzhledem k potřebě programu pracovat s komplexními maticemi, je pro zacházení s komplexními čísly použita externí knihovna NumPy, která bude detailněji popsána v samostatné kapitole.

### 3.3 Knihovny jazyka Python

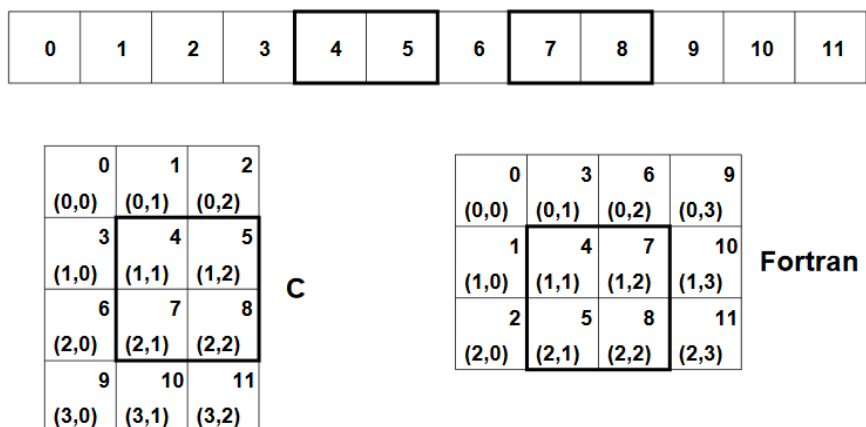
Mezi přední výhody Pythonu patří množství dostupných knihoven. Nezávisle na vývoji samotného jazyka vzniká řada knihoven doplňující jeho funkcionalitu ve vztahu ke konkrétním odvětvím. V této kapitole budou uvedeny knihovny přímo použité při realizaci vyvíjeného programu.

#### 3.3.1 Knihovna NumPy

NumPy je knihovna umožňující práci s velkými vícedimenzionálními poli a maticemi. Jedná se o výchozí knihovnu pro vědeckou a analytickou činnost. Její počátky se datují do roku 2005 a je vyvíjena v rámci open source licence. K datu psaní práce je vydávána v stabilní verzi 1.14.0.

Fundamentálním objektem knihovny je N-dimenzionální pole (`ndarray`). Toto pole je homogenní uspořádání prvků indexovaných použitím  $N$  čísel. Každé pole je definováno pomocí dvou důležitých parametrů: 1) tvar pole a 2) typ prvků tvořící toto pole. Tvar pole je  $n$ -tice (neměnitelný seznam) o  $N$  číslech (pro každou dimenzi), která udává rozsah změny indexu v dané dimenzi. Každý prvek pole zabírá fixní počet bytů a je jím většinou číslo. Prvek ale může být libovolného datového typu. Této flexibility je docíleno použitím objektu `data-type` (`dtype`). Ten je přiřazen každému vytvořenému poli a zároveň má přiřazený jeden ze základních datových typů Pythonu. V NumPy je 21 různých dtypů a každý z nich je schopen pojmout referenci na všechny ostatní typy. Tím je docíleno flexibilního použití datových typů v polích.

Z hlediska práce s pamětí je N-dimenzionální pole pouze jedno-dimenzionální sekvence s jedno-dimenzionálním indexem, který je nutnou podmínkou pro správnou adresaci paměti. V práci je v matematické části počítáno s maticemi. Ty nejsou v NumPy samostatným objektem, ale jedná se o dvoudimenzionální případ N-dimenzionálního pole. V paměťovém rozložení `ndarray` podporuje jak C tak Fortran indexování. Propojeny jsou pak pomocí transpoziční operace. Pro ilustraci je níže uveden (Obrázek 3.2) způsob rozložení pro dvoudimenzionální pole.



Obrázek 3.2 Paměťové rozložení C a Fortran

Výpočetní rychlost NumPy v porovnání s ostatními komerčně používanými jazyky (MATLAB, Mathematica) závisí silně na způsobu používání knihovny. Tady si je nutné uvědomit, že NumPy řadu úkonů řeší voláním skrytých funkcí psaných v C, C++ nebo Fortranu. Platí tedy, že nejrychlejšího výpočtu dosáhneme při aplikaci jedné funkce na celé pole raději než při aplikaci této funkce postupně na jednotlivé elementy pole stejného. Pro ilustraci mějme dvoudimenzionální pole  $A$  o rozměru  $10 \times 20$ , u kterého chceme zvětšit všechny prvky prvního řádku o jedna. V níže uvedeném kódu nejprve vytvoříme komplexní matici Fortranového indexování. Následně měříme dobu výpočtu, kdy pomocí cyklu postupně zvětšujeme prvky prvního řádku, a dobu výpočtu kdy toto číslo přičteme k výřezu (`slicing`) z matice jako celku. Jako výsledek bereme nejrychlejší průběh ze sedmi realizovaných běhů.

Z níže uvedené ukázky (Ukázka kódu 3.2) je zřejmé, že aplikace operace na výřez je více než dvakrát rychlejší. Pro návrh programu z toho plyne důležitý závěr. Při práci s maticemi je důležité vyvarovat se použití Python cyklů a operace vždy provádět na celku co největších rozměrů.

```
import timeit

#Vytvoření pole
setup = '''from numpy import matlib
A = matlib.zeros((10, 20), dtype = complex, order = "F")'''

#Časování výpočtu s použitím cyklů
print(min(timeit.Timer('for x in range (0,A.shape[1]):A[1,x] += 1', setup =
setup).repeat(7, 1000)))

#Časování výpočtu s použitím slicing
print(min(timeit.Timer('A[1,:] += 1', setup = setup).repeat(7, 1000)))

#Výstup
0.048088619590360204
0.023481298191374067
```

*Ukázka kódu 3.2 Měření výpočetní doby operací na NumPy poli*

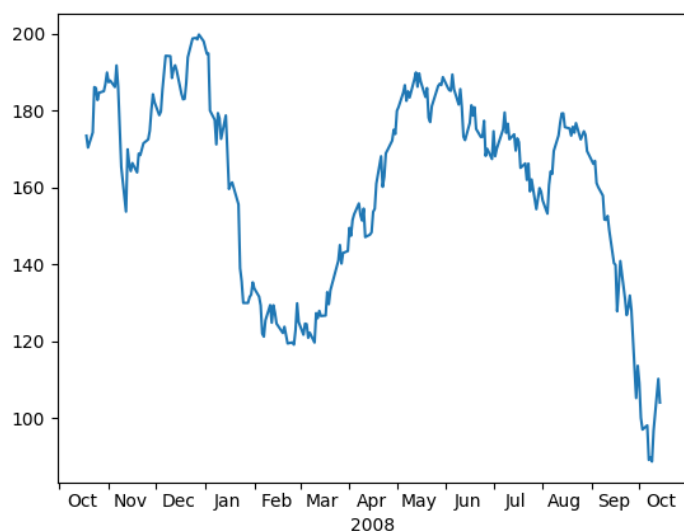
### 3.3.2 Knihovna SciPy

Scipy je knihovna určená pro vědecké a technické výpočty. Je součástí takzvaného NumPy stack spolu s Matplotlib, pandas a SymPy. V práci jsou z jejich modulů využity především interpolační nástroje.

Knihovna nabízí řadu jednorozměrných i vícerozměrných interpolací. Jelikož jsou v práci aproximovány pouze materiálové parametry vrstev, je použito výhradně interpolace pomocí jednorozměrné spline křivky.

### 3.3.3 Knihovna Matplotlib

Matplotlib je knihovna určená k vykreslování dat. Původně vznikla jako emulace vykreslování v jazyku MATLAB a dodnes je mu v řadě ohledů podobná. Ve výpočetní části je navázána na knihovnu NumPy a obsahuje API pro integraci v grafických knihovnách. Základní filosofií je dle autora John D. Huntera důraz na jednoduchost a přehlednost (Obrázek 3.3)



*Obrázek 3.3 Ilustrační graf Matplotlib*

### 3.3.4 Knihovna Tkinter

Tkinter je knihovna určená k tvorbě grafických rozhraní. Je součástí standardní instalace Pythonu. Ačkoliv je stále jednou z nejpoužívanějších grafických knihoven, zaostává funkcionalitou za dnes již pokročilejšími knihovnami jako PyQt, wxPython či Kivy. Tato práce neklade nárok na možnosti grafického rozhraní a v práci je knihovna Tkinter použita pro jednoduchost její integrace.

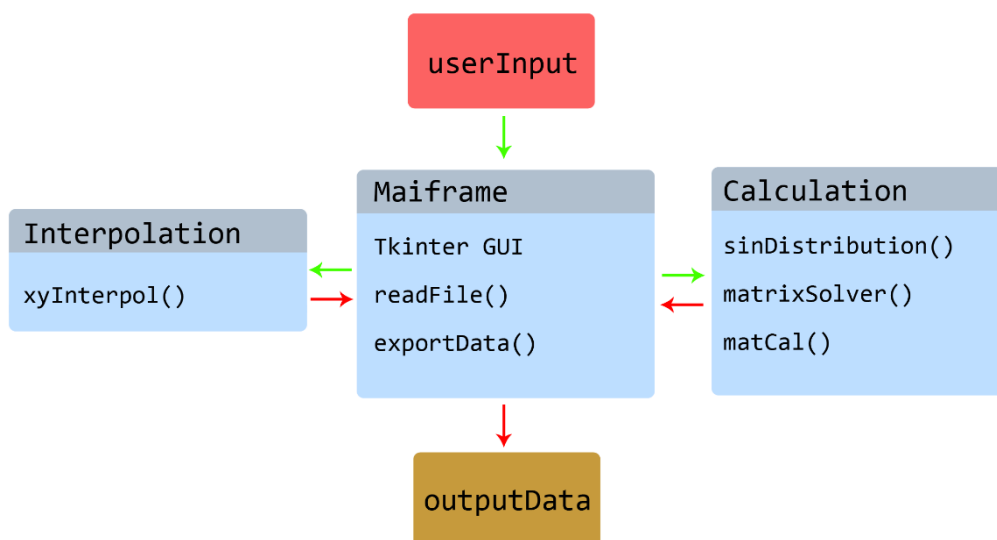
## 4 Vývoj samotného programu R Lay

R Lay je program vyvíjený v rámci této diplomové práce. Jeho hlavní funkcionalitou je výpočet chování světla v definovaných, především tenkovrstvých, strukturách. Oproti základnímu výpočtu umožňuje program zahrnutí difuzních rovin a následné řešení šíření rozptýleného světla.

Veškerý zdrojový kód je realizován v jazyce Python verze 3.6 s použitím externích knihoven. Výsledný program lze nativně spouštět na Linuxových distribucích. Pro Windows je možné program převést pomocí py2exe na spustitelný soubor Microsoft Visual C runtime, nebo distribuovat spolu s instalací Pythonu. Komunikace s uživatelem probíhá v anglickém jazyce, a tedy i jednotlivé ukázky uživatelského rozhraní mají anglické popisky.

### 4.1 Struktura programu

Program lze dle funkcionalit rozdělit do tří hlavních bloků (Obrázek 4.1): Mainframe, DataManagement a Calculation. Část Mainframe představuje základní kostru programu. Obsahuje grafické uživatelské rozhraní (GUI) a s ním spojené funkce pro import a export dat. Jejím výstupem jsou parametry modelované struktury a její materiálové údaje. DataManagement je blok zajišťující převod vstupních dat na spojitě funkce a vyhlazování případných anomálií průběhů. Calculation je poté část realizující samotný výpočet.



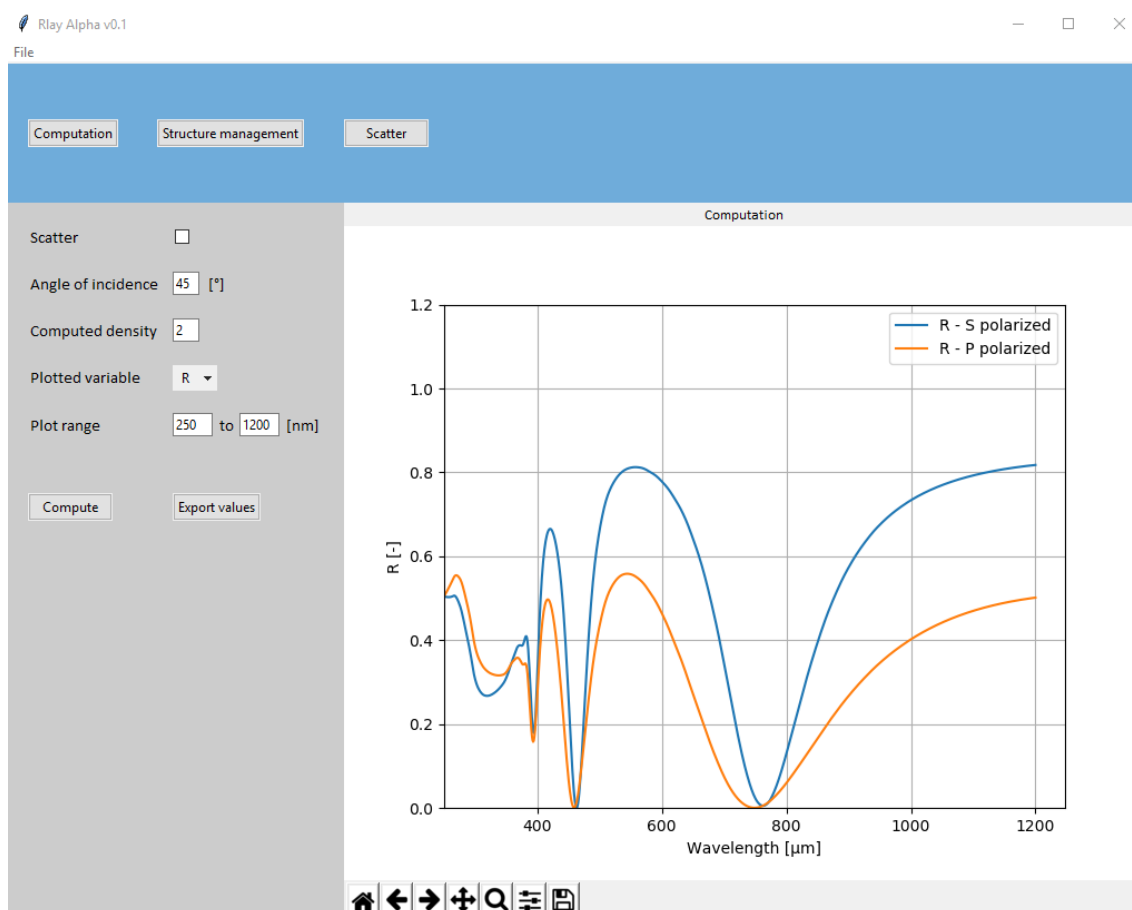
Obrázek 4.1 Struktura programu R Lay

## 4.2 Modul Mainframe

Základ celého programu tvoří GUI vytvořené pomocí nativní Python knihovny Tkinter. Vzhledem k relativně nízké komplexitě GUI je aplikace realizována jako instance hlavní třídy *Mainframe*, která dědí ze základní Tkinter třídy *Tk*. Tento způsob má při menších rozměrech programu výhodu v jednodušší organizaci kódu. Konceptně se poté program skládá ze tří hlavních oken se společnou navigační částí.

Prvním oknem je *Computation*. V této části uživatel spouští samotný výpočet a nastavuje tyto obecné parametry (Obrázek 4.2):

- zahrnutí rozptylu (*Scatter*);
- uhel dopadajícího světla (*Angle of incidence*);
- hustota výpočtu (*Computed density*);
- vykreslené veličiny (*Plotted variable*);
- rozsah výpočtu (*Plot range*).



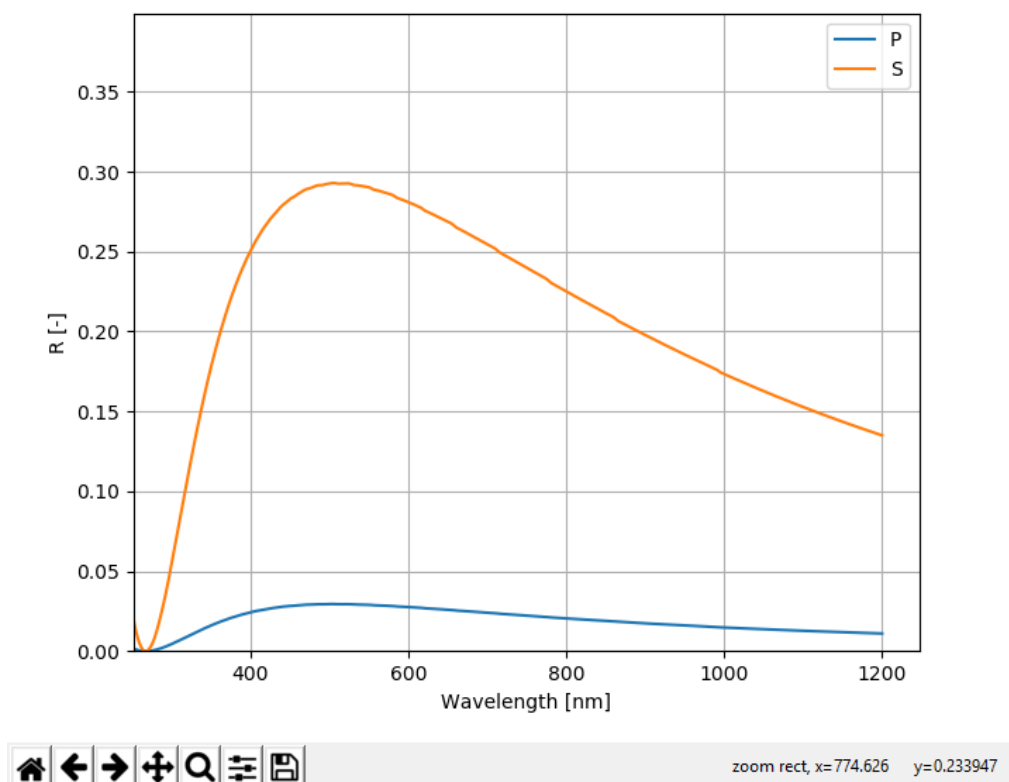
Obrázek 4.2 Okno *Computation* programu *RLay*

Úhel dopadajícího světla pro výpočet představuje úhel, v jakém dopadá na strukturu světlo s intenzitou rovno jedné. Hodnota úhlu tvoří následně základ pro vytvoření úhlového vektoru, který bude podrobněji popsán v části *Calculation*. Zahrnutí difuze se provádí nastavením difuzních rovin v oknu *Scatter*. Parametr hustota výpočtu souvisí se zahrnutím těchto rovin. Při výpočtu, kdy je pomyslný paprsek světla



difundován do více směrů na určitých rovinách, je nutné výpočet rozšířit o matice se schopností tento úhlový vektor popsat. Jelikož počet úhlů v tomto vektoru je teoreticky nekonečný, hustota výpočtu popisuje množství diskrétních úhlů, pro které je šíření strukturou počítáno. Difundované světlo je do těchto úhlu ve výpočtu „rozdělováno“. Platí tedy, že s rostoucí hustotou výpočtu roste i přesnost výpočtu difundovaného světla. Zároveň ale lineárně roste komplexnost, a tedy i výpočetní čas.

Parametr vykreslené veličiny dává uživateli možnost zadat, jaké veličiny budou zobrazeny ve výstupním grafu. Program pro danou strukturu počítá absorpci, reflexi a transmisi pro obě polarizace světla. Do grafu je možné zahrnout libovolnou kombinaci těchto veličin. Rozsah simulace je zadáván v nanometrech. Pro vykreslení grafu je použito knihovny Matplotlib, která mimo standardních nástrojů pro tvorbu grafů obsahuje i nástavbu pro uživatelskou manipulaci výstupu. Uživateli je umožněno zobrazení vybraných úseků grafu, navigování mezi provedenými kroky, ukládání do libovolného výstupního grafického formátu a zobrazení aktuální pozice kurzoru v  $xy$  souřadnicích. Příklad výstupního grafu reflexe (Obrázek 4.3) je uveden pro světlo dopadající pod úhlem  $45^\circ$  na vrstvu oxidu křemičitého o tloušťce 100 nm.

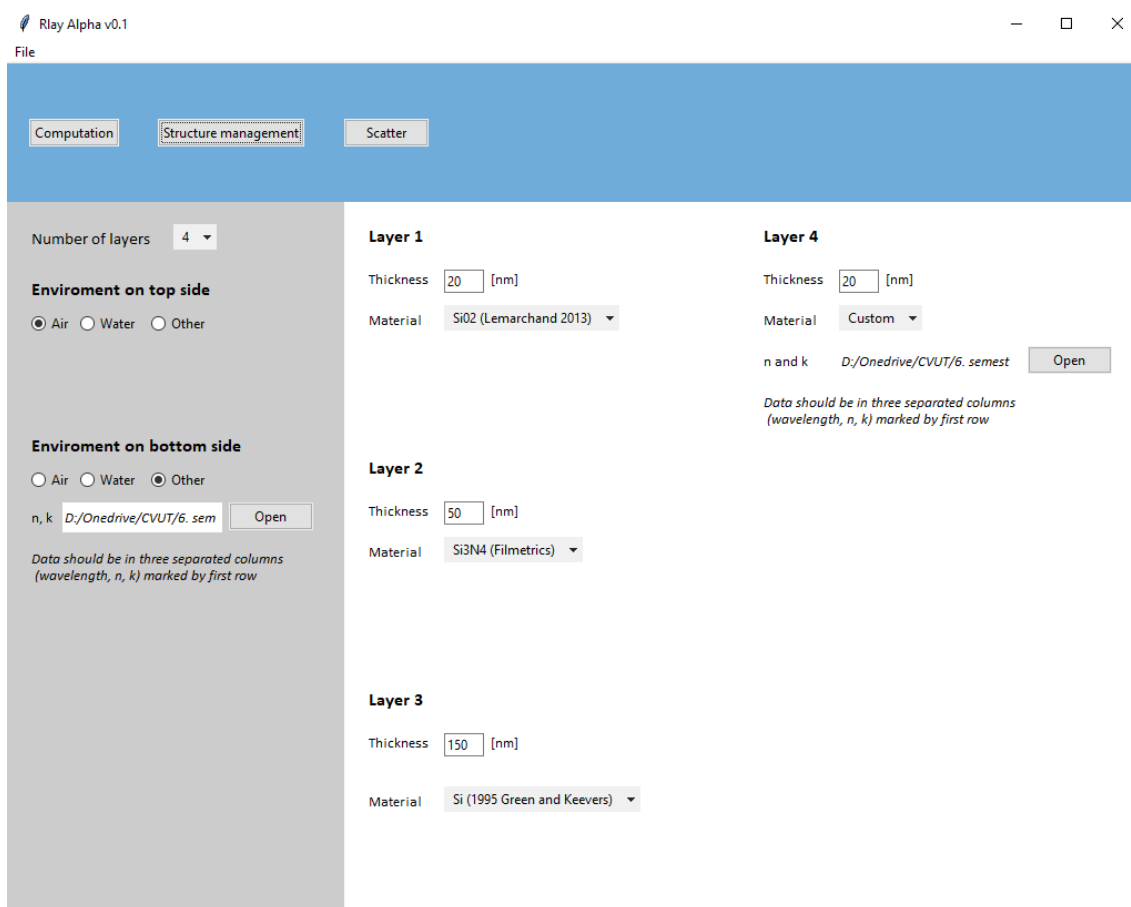


Obrázek 4.3 Příklad výstupního grafu programu RLay

Druhé okno Structure management umožňuje návrh simulované struktury (Obrázek 4.4). V levé části je možné zvolit počet vrstev a materiál prostředí na jednotlivých stranách struktury. Prostředí lze zvolit z předem definovaných materiálů (vzduch, voda) nebo může uživatel nahrát soubor s indexem lomu a extinkčním koeficientem. Pro zvolený počet vrstev je poté možné zvolit tloušťku v nanometrech a obdobně jako u prostředí lze jejich materiál vybrat z předem definovaného seznamu nebo nahrát z externího souboru. V programu jsou přítomny následující předvolené materiály pro vrstvy:

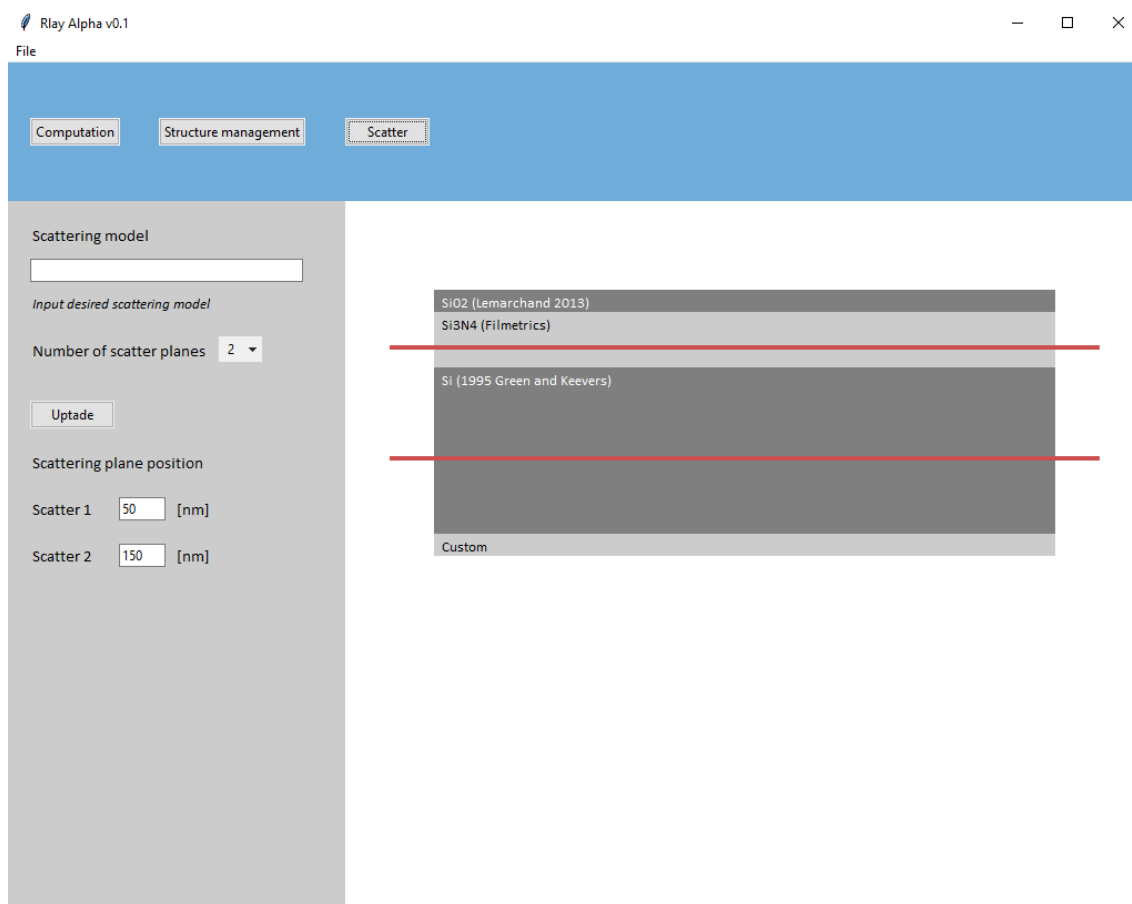
*Tabulka 4.1 Zabudované materiály programu RLay*

Materiál	Autor	Rozsah [nm]
Si	Green and Keevers 1995 [7]	250 - 1 450
CdTe	Treharne 2011 [8]	300 - 1500
Al	Rakic 1995 [9]	0,124 - 200 000
Al <sub>2</sub> O <sub>3</sub>	Querry 1985 [10]	210 - 12 500
SiO <sub>2</sub>	Lemarchard 2013 [11]	250 - 2 500
ZnO	Stelling 2017 [12]	302 - 1 685
Si <sub>3</sub> N <sub>4</sub>	Filmterics [13]	180 - 2 500



*Obrázek 4.4 Okno Structure programu RLay*

Poslední okno Scatter (Obrázek 4.5) umožňuje v levé části analytické zadání difuzního modelu a umístění difuzních rovin. Pozice rovin je zadávána jako vzdálenost v nanometrech od vrchní strany struktury. Pro navrženou strukturu je v pravé části možnost zobrazit vizualizaci struktury s příslušným umístěním difuzních rovin a názvem materiálu jednotlivých vrstev. Vrstvy jsou střídavě zbarveny pro jednodušší rozeznání jejich rozhraní.



**Obrázek 4.5** Okno Scatter programu Rlay

Poslední důležitou součástí bloku Mainframe jsou funkce pro import a export dat. Pro import lze data nahrávat ze souborů csv a txt s hodnotami oddělenými tabulátory nebo čárkou. Pro materiálové parametry jsou očekávány sloupce: vlnová délka ( $w$ ), index lomu ( $n$ ) a extinkční koeficient ( $\kappa$ ). Výstupní hodnoty jsou exportovány do standardního formátu .csv.

### 4.3 Modul DataManagment

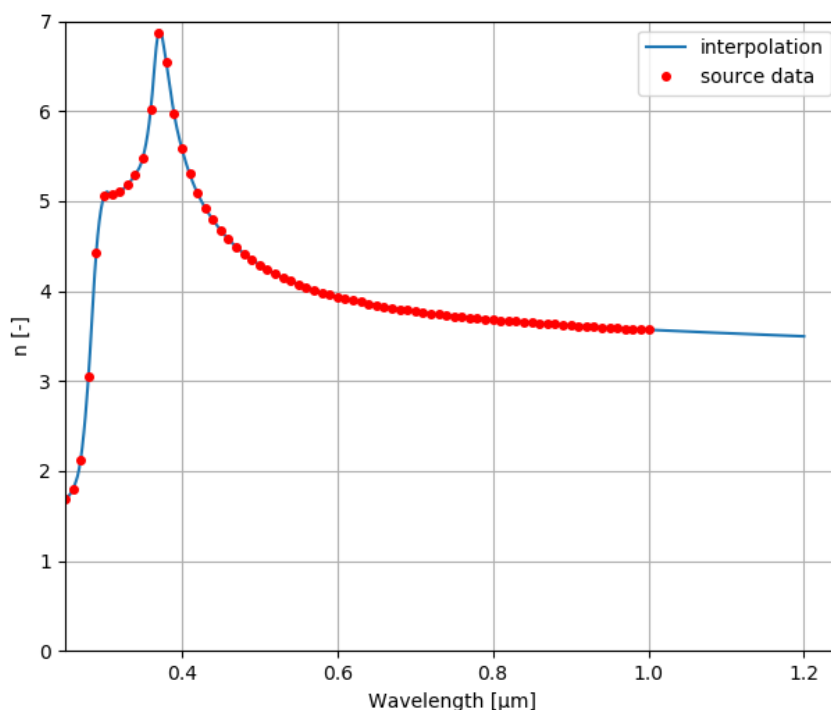
Modul DataManagment je zodpovědný za zpracování vstupních dat. Těmi jsou hodnoty extinkčního koeficientu a indexu lomu pro jednotlivé vlnové délky. K umožnění výpočtu pro libovolné vlnové délky v zadaném rozsahu jsou vstupní hodnoty interpolovány.

Pro interpolaci je využit modul `interpolate` knihovny SciPy [14]. Jako typ interpolace je použito `UnivariateSpline`, tedy interpolace postupným polynomem (Ukázka kódu 4.1). Jednotlivé úseky jsou aproximovány polynomy s podmínkou, že derivace těchto polynomů je v navazujících bodech (knots) identická. Počet těchto bodů lze měnit parametrem (`k`) a tím je umožněno vyhlazení výstupní křivky.

```
def xyInterpol(x, y, order):  
  
    xi = np.array(x)  
    yi = np.array(y)  
  
    s = InterpolatedUnivariateSpline(xi, yi, k=order)  
  
    return (s)
```

#### *Ukázka kódu 4.1 Interpolace vstupních dat*

Pro ilustraci je níže (Obrázek 4.6) uveden graf porovnání interpolační křivky (interpolation) a zdrojových dat (source data). Zvoleným materiálem je křemík s daty publikovanými dvojicí Green and Keevers v roce 1995 [7]. Interpolace tato data reprodukuje téměř identicky a je schopna poskytnout data i pro vlnové délky vyšší než rozsah dat zdrojových (0 – 1000 nm).



*Obrázek 4.6 Interpolace zdrojových dat*

## 4.4 Modul Calculation

Modul Calculation je klíčovou částí programu RLayer. Probíhá v něm samotný výpočet chování světla v definované struktuře. Pro výpočet je využito maticové metody popsané v kapitole 2.6. Pro danou strukturu definovanou uživatelem přes grafické uživatelské rozhraní je sestavena rozptylová matice  $\mathbf{S}$  tuto strukturu charakterizující.

### 4.4.1 Polarizace vlny

Z rovnice 2.32 víme, že pro strukturu o  $M$  vrstvách je tato matice definována jako součin jednotlivých matic rozhraní  $\mathbf{I}^{(ab)}$  a matic vrstev  $\mathbf{L}^{(m)}$ :

$$\mathbf{S} = \mathbf{I}^{(01)}\mathbf{L}^{(1)}\mathbf{I}^{(12)}\mathbf{L}^{(2)} \dots \mathbf{L}^{(M)}\mathbf{I}^{(M,M+1)} \quad (4.1)$$

V případě popsaném v teoretické části má rozptylová matice  $\mathbf{S}$  rozměr  $2 \times 2$  a popisuje vztah mezi dopadající, prošlou a odraženou vlnou jedné polarizace pro definovanou strukturu. V realitě je případ, kdy na strukturu dopadá polarizovaná vlna poměrně vzácný a mnohem častěji se setkáme s vlnou nepolarizovanou. Rozptylová matice  $\mathbf{S}$  musí tedy obsahovat prvky pro S i P-polarizovanou vlnu. Vlnový vektor pravé strany má poté podobu:

$$\begin{pmatrix} \varepsilon_{as}^+ \\ \varepsilon_{as}^- \\ \varepsilon_{ap}^+ \\ \varepsilon_{ap}^- \end{pmatrix} \quad (4.2)$$

kde spodní index  $s, p$  značí příslušnost k jednotlivým polarizacím. Jelikož se obě polarizace nepromíchávají, jedná se o nezávislé složky. Rozptylová matice  $\mathbf{S}$  je tedy maticí rozměru  $4 \times 4$  s diagonálně umístěnými maticemi  $2 \times 2$  pro jednotlivé polarizace:

$$\mathbf{S} = \begin{pmatrix} \mathbf{S}_s & 0_{2 \times 2} \\ 0_{2 \times 2} & \mathbf{S}_p \end{pmatrix} \quad (4.3)$$

kde  $\mathbf{S}_s$  je rozptylová matice pro S-polarizovanou vlnu,  $\mathbf{S}_p$  je rozptylová matice pro P-polarizovanou vlnu a  $0_{2 \times 2}$  je nulová matice o rozměru  $2 \times 2$ . Obdobně lze vztah pro rozptylovou matici  $\mathbf{S}$  (4.1) definovat pomocí matic rozhraní pro S-polarizovanou vlnu  $\mathbf{I}_s^{(ab)}$ , matic rozhraní pro P-polarizovanou vlnu  $\mathbf{I}_p^{(ab)}$ , matic vrstev pro S-polarizovanou vlnu  $\mathbf{L}_s^{(m)}$  a matic vrstev pro P-polarizovanou vlnu  $\mathbf{L}_p^{(m)}$ :

$$\mathbf{S} = \begin{pmatrix} \mathbf{I}_s^{(01)} & 0_{2 \times 2} \\ 0_{2 \times 2} & \mathbf{I}_p^{(01)} \end{pmatrix} \begin{pmatrix} \mathbf{L}_s^{(1)} & 0_{2 \times 2} \\ 0_{2 \times 2} & \mathbf{L}_p^{(1)} \end{pmatrix} \dots \begin{pmatrix} \mathbf{L}_s^{(M)} & 0_{2 \times 2} \\ 0_{2 \times 2} & \mathbf{L}_p^{(M)} \end{pmatrix} \begin{pmatrix} \mathbf{I}_s^{(M,M+1)} & 0_{2 \times 2} \\ 0_{2 \times 2} & \mathbf{I}_p^{(M,M+1)} \end{pmatrix} \quad (4.4)$$

## 4.4.2 Úhlový vektor

Jednou z klíčových funkcionalit programu RLay je možnost zahrnout do výpočtu difuzi světla. Před popisem metody výpočtu difuze ve struktuře je nejprve vhodné popsat změnu tvaru výsledné matice  $\mathbf{S}$  při výpočtu s difuzí světla.

Difuzi světla ve vrstvě či rozhraní lze definovat jako místo vzniku slabších sekundárních vln s úhlem šíření jiným než vlna, která je způsobila. V kapitole 2.8 byly prvky matice rozhraní  $\mathbf{I}^{(ab)}$  a vrstev  $\mathbf{L}^{(m)}$  definovány jako závislé na úhlu dopadající nebo lomené vlny k normále. Prvky rozptylové matice  $\mathbf{S}$  jsou tedy také na úhlu závislé a nelze jimi popsat světlo difundované do úhlů jiných. Jelikož je teoretický počet úhlů, do kterých vlna difunduje, nekonečný, musel by i počet rozptylových matic  $\mathbf{S}$  být nekonečný. Program RLay obsahuje uživatelský parametr hustota výpočtu (Computed density), který udává množství úhlů, pro které výpočet proběhne.

O vytvoření výsledného úhlového vektoru se stará funkce `sinDistribution`. Na první pohled by se jako evidentní cesta mohlo jevit rozdělení úhlu dopadu vlny  $\theta_i$  rovnoměrně mezi hodnoty 0 a 90 ° dle hustoty výpočtu. Tento způsob by ale byl funkční pouze pro rozptyl pro konstantní index lomu. Vyjdeme-li ze Snellova zákona, můžeme pro součin komplexního indexu lomu vrstvy  $N^{(m)}$  a úhlu šíření v této vrstvě  $\theta^{(m)}$  psát vztah:

$$N^{(1)} \sin \theta^{(1)} = N^{(2)} \sin \theta^{(2)} = \dots = N^{(M)} \sin \theta^{(M)} \quad (4.5)$$

kde  $M$  je celkový počet vrstev dané struktury. Pokud nyní definujeme komplexní indexy lomu struktury dvou vrstev tak, že druhá vrstva má reálnou část tohoto indexu vyšší než ta předešlá  $N^{(1)} < N^{(2)}$ , musí pro siny úhlů šíření platit:

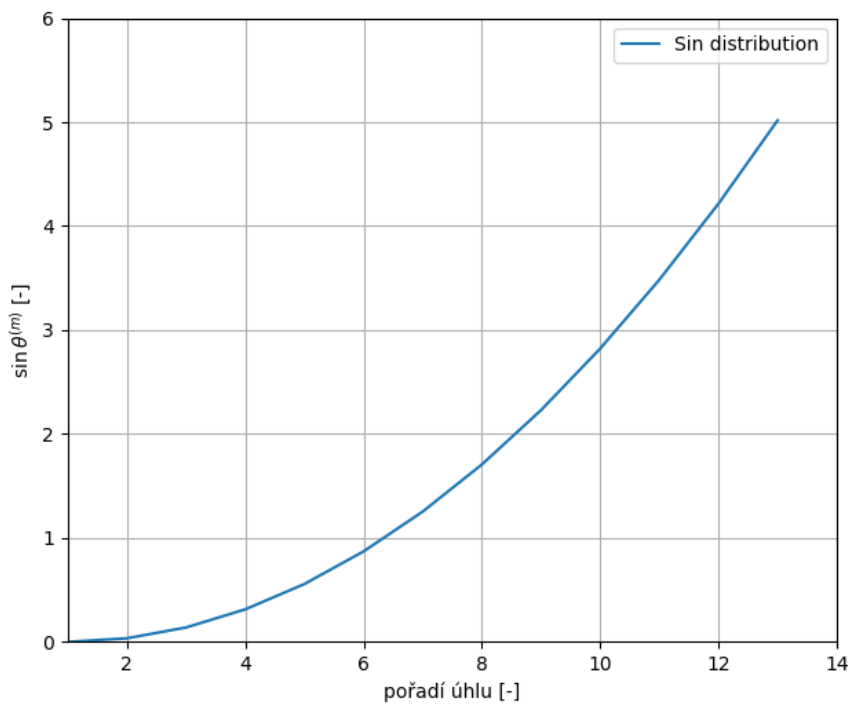
$$\sin \theta^{(1)} > \sin \theta^{(2)} \quad (4.6)$$

Ve vrstvě druhé, tak dle rovnice 4.6 je vektor úhlů  $\{0, \dots, \sin \theta^{(2)}\}$  výrazně užší a nemůže dojít k difuzi do některých úhlů. V našem výpočtu proto používáme následující postup. Pro zjednodušení výpočtu je výhodné diskretizovat veličinu  $\sin \theta^{(m)}$ , protože její hodnota se dle rovnice 4.5 zachovává. Zároveň se můžeme vyhnout diskretizaci komplexní veličiny, dovolíme-li veličině  $\sin \theta^{(m)}$  nabývat hodnoty větší než jedna, protože poté zůstanou hodnoty  $\theta^{(m)}$  a  $\cos \theta^{(m)}$  komplexní. K tomu aby zároveň hodnoty  $\sin \theta^{(m)}$  byly skutečně reálné, učiníme aproximaci, kdy pro Snellův zákon budeme počítat pouze reálné části indexů lomu.

Pro zvolení krajního bodu tohoto vektoru lze vycházet z nejvyššího naměřeného indexu lomu přírodního materiálu Germania, který v infračervené oblasti dosahuje reálné části indexu lomu  $n = 4,13$  [15]. V programu RLay je proto jako okraj intervalu zvolen úhel  $\theta^{(n)}$ , pro který platí:

$$n^{(0)} \sin \theta^{(n)} = 5 \quad (4.7)$$

kde  $n^{(0)}$  je reálná část indexu lomu vstupního prostředí,  $\sin \theta^{(n)}$  horní hranice vstupního úhlového vektoru a  $n$  počet úhlů v úhlovém vektoru. Z definice sinu víme, že pro reálné úhly nabývá hodnot v rozmezí  $(0, \dots, 1)$ . Siny vyšší než jedna jsou pro úhly ryze imaginární. Tato skutečnost je klíčová při uvažování evanescentní vlny uvnitř struktury. Evanescentní vlna vzniká na rozhraní prostředí při totální reflexi. Vlna tohoto typu může přispívat k následné difuzi a propagaci vlny. Diskretizací úhlového vektoru jsme schopni její interakci ve struktuře popsat.



**Obrázek 4.7** Rozložení sinu úhlu v úhlovém vektoru pro 13 prvků

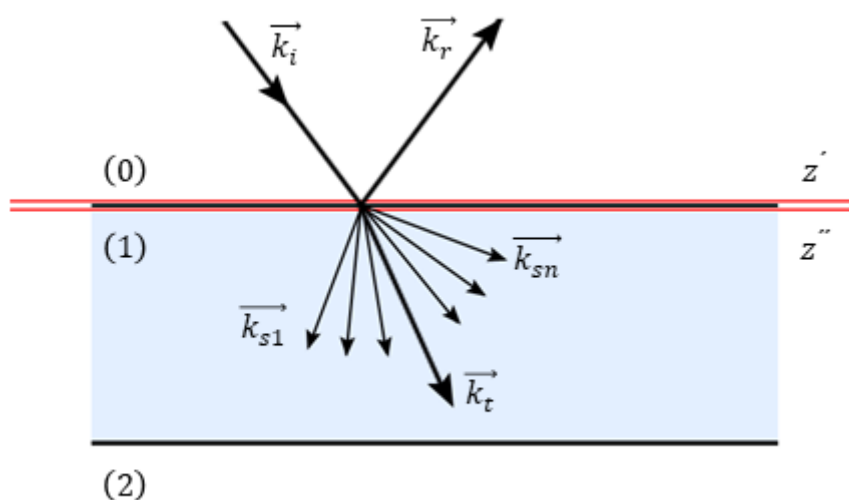
Prvky výsledného vektoru úhlů jsou vázány na uživatelem zadanou hustotu výpočtu. Při jeho nejnižší hodnotě obsahuje vektor úhlů tři prvky sinu úhlu. Krajní hodnoty jsou pevně stanoveny na  $\{0, \dots, 5\}$ . Třetí hodnotou je vždy sinus uživatelem zadaného úhlu dopadu vlny na strukturu  $\theta_i$ . Toto je nezbytné, jelikož úhel dopadu bude i po zahrnutí difuze představovat cestu šíření majoritní části vlny ve struktuře. Jeho případnou aproximací do krajních bodů intervalu, který by ho obsahoval, by docházelo k nepřiměřenému nárůstu chyby výpočtu. Zbylé hodnoty jsou v intervalu rozmístěny parabolickou funkcí:

$$y(x) = \left( \frac{9}{4n} (x - 1) \right)^2 \quad (4.8)$$

kde  $n$  je celkový počet úhlů ve vektoru. Touto funkcí je docíleno hustšího rozložení v reálné oblasti ( $0 - 90^\circ$ ) a naopak řidšího v oblasti imaginárních úhlů ( $\sin \theta > 1$ ). Pro ilustraci je níže zobrazena funkce (Obrázek 4.7) pro případ, kdy úhlový vektor obsahuje 13 prvků.

### 4.4.3 Difuze ve struktuře

Z matematického hlediska je vhodné, aby vlna vzniklá difuzí v počítané struktuře byla počítána nezávisle na vlně bez difuze světla. Vlna vzniklá difuzí je přitom reprezentována stejným vektorem úhlů a polarizace.



Obrázek 4.8 Difuze vlny na rozhraní

Difuze ve struktuře je popsána difuzní maticí  $\mathbf{D}$ . Nejprve je vhodné definovat její místo v součinu tvořící rozptylovou matici  $\mathbf{S}$ , které může být dvojího typu. Při prvním z nich mějme difuzní rovinu umístěnou na rozhraní prostředí (0) a struktury o jedné vrstvě (1) (Obrázek 4.8). V tomto případě musí být difuzní matice přítomna na obou stranách matice rozhraní  $\mathbf{I}^{(01)}$  v rovinách  $z'$  a  $z''$ , které se limitně blíží rozhraní. Aby bylo možné docílit přerozdělení prvků matice  $\mathbf{S}$ , musí být difuzní matice z matematického hlediska vždy násobeno zleva. Vztah pro výslednou rozptylovou matici lze zapsat:

$$\mathbf{S} = \mathbf{D}_2^{(01)} \mathbf{D}_1^{(01)} \mathbf{I}^{(01)} \mathbf{L}^{(1)} \mathbf{I}^{(12)} \quad (4.9)$$

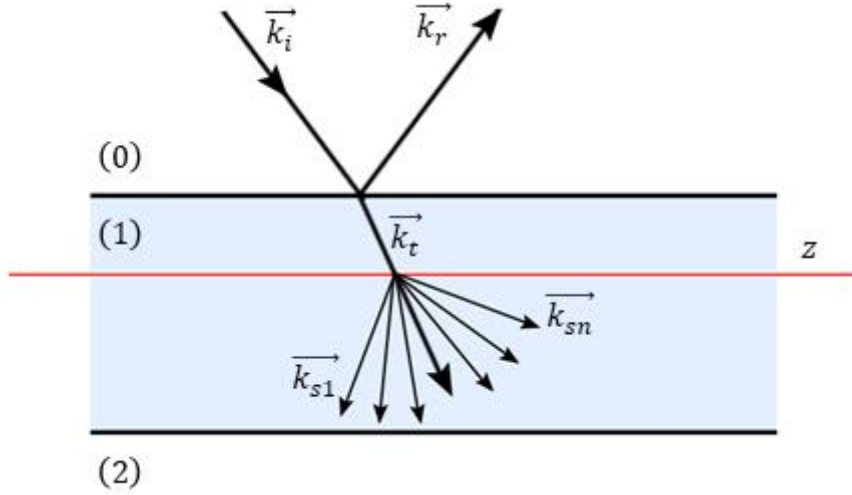
kde  $\mathbf{D}_1^{(01)}$  je difuzní matice na straně k prostředí (0) a  $\mathbf{D}_2^{(01)}$  na straně k vrstvě (1).

Druhým případem je difuze vlny uvnitř vrstvy. Oproti minulému případu stačí k popsání jevu pouze jedna difuzní matice  $\mathbf{D}$  umístěna na rovině  $z$  uvnitř vrstvy (1) (Obrázek 4.9). Vztah pro výslednou rozptylovou matici dostává tvar:

$$\mathbf{S} = \mathbf{D}^{(1)} \mathbf{I}^{(01)} \mathbf{L}_1^{(1)} \mathbf{L}_2^{(1)} \mathbf{I}^{(12)} \quad (4.10)$$



kde  $\mathbf{D}^{(1)}$  je difuzní matice ve vrstvě (0) a  $\mathbf{L}_1^{(1)}, \mathbf{L}_2^{(1)}$  jsou části vrstvy (1) rozdělené rovinou  $z$ .



Obrázek 4.9 Difuze vlny ve vrstvě

Prvky difuzní matice  $\mathbf{D}$  musí zajistit, aby vlna interagující se strukturou byla zeslabena ve svém směru šíření a zeslabená část rozložena do jednotlivých prvků úhlového vektoru. Pro prvky matice rozptýleného světla lze psát:

$$D_{ij} = s(\lambda n \sigma) \delta_{ij} \mathbf{U}_{4 \times 4} + ADF(\lambda n \sigma \theta_i) [1 - s(\lambda n \sigma)] \mathbf{W}_{4 \times 4} \quad (4.11)$$

kde  $\mathbf{U}_{4 \times 4}$  a  $\mathbf{W}_{4 \times 4}$  představují matice:

$$\mathbf{U}_{4 \times 4} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.12)$$

$$\mathbf{W}_{4 \times 4} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \quad (4.13)$$

Matice  $\mathbf{W}_{4 \times 4}$  není diagonální z důvodu ztráty polarizace při difuzi světla. Světlo jedné polarizace se při difuzi rovnoměrně rozdělí mezi obě polarizace. Parametr  $s(\lambda n \sigma)$  představuje difuzní faktor a nabývá hodnot  $0 < s(\lambda n \sigma) < 1$ .  $\delta_{ij}$  představuje Kronekovo delta. Parametr  $\sigma$  představuje hrubost rozptylové roviny. Dále je rozptyl závislý na indexu lomu a na vlnové délce, viz [16].  $ADF(\lambda n \sigma \theta_i)$  je úhlové rozložení difundovaného světla. V každém místě struktury přitom může dojít pouze k difuzi do reálných úhlů úhlového vektoru v daném místě a musí platit:

$$\sum_{j=1}^M ADF^2(\lambda n \sigma \theta_{ij}) = 1 \quad (4.14)$$

kde  $M$  je počet úhlů do kterých je vlna rozptýlena. Funkce  $ADF(\lambda n \sigma \theta_i)$  je do programu zadávána přes textové pole uživatelského rozhraní

#### 4.4.4 Výsledná rozptylová matice

S předchozích kapitol víme, že oproti případu odvozeném v kapitole 2.6 musí výsledná rozptylová matice obsahovat navíc matice pro jednotlivé druhy polarizace, difundované světlo a ostatní prvky úhlového vektoru mimo úhel dopadající vlny  $\theta_i$ . Rozměr této matice  $\mathbf{S}_{n \times n}$  můžeme v závislosti na počtu prvku úhlového vektoru zapsat jako:

$$n = 2n_v(n_s + n_p) \quad (4.15)$$

kde  $n_v$  je počet prvků úhlového vektoru,  $n_s$  je rozměr rozptylové matice pro S-polarizovanou vlnu a  $n_p$  je rozměr rozptylové matice pro P-polarizovanou vlnu. Z kapitoly 2.6 víme, že rozměr rozptylové matice pro jednotlivou polarizaci je vždy  $2 \times 2$ . Proto vztah pro rozměr výsledné rozptylové matice (4.15) můžeme zjednodušit:

$$n = 2n_v(2 + 2) = 8n_v \quad (4.16)$$

`class Layer:`

```

def __init__(self, thickness, N, wavelength):
    self.thickness = thickness
    self.N = N
    self.wavelength = wavelength

def getResult(self, data):
    sinMatrix = data[0][0]
    S = data[1]
    M = data[2]

    L = matlib.zeros((8 * len(sinMatrix), 8 * len(sinMatrix)),
                    dtype=complex, order="F")

    for l in range(0, len(sinMatrix)): ...

    return ([sinMatrix, data[0][1], data[0][2]], S, M)

```

*Ukázka kódu 4.2 Třída Layer výpočetní funkce*

Z programového hlediska jsou jednotlivé matice součinu tvořícího výslednou rozptylovou matici reprezentovány jako objekty. Výpočetní část tedy obsahuje definované třídy pro rozhraní `Interface`, pro vrstvu `Layer` a pro difuzi `Difuse`. Všechny tyto třídy obsahují metodu `getResult()` (Ukázka kódu 4.2).

Podoba této metody je dána postupem výpočtu v programu. Jako první je při zavolání výpočetní funkce ze vstupních parametrů vytvořen seznam (`list`) obsahující objekty jednotlivých tříd v pořadí v jakém jsou v součinu tvořícím rozptylovou matici. Následně je pomocí cyklu pro každý objekt postupně zavolána metoda `getResult()`, která očekává parametry `self` a `data`. První z parametrů je v Pythonu konvenčně používán jako styl ukazování na vlastní parametry instance. Druhý parametr obsahuje data předávaná mezi objekty při výpočtu. Jedná se o seznam a má následující prvky:

*Tabulka 4.2 Parametr data maticových objektů*

Pozice	Popis
<code>data[0][0]</code>	úhlový vektor
<code>data[0][1]</code>	vektor vln levé strany
<code>data[0][1]</code>	vektor vln pravé strany
<code>data[1]</code>	rozptylová matice v místě výpočtu
<code>data[2]</code>	seznam předchozích matic

Samotný výpočet nejen postupně vyhodnocuje součin jednotlivých matic (4.1), ale také zachovává v paměti jednotlivé mezikroky součinu a dílčí matice. Výpočet je mimo řešení součinu připraven na řešení dílčích částí s minimálním dopadem na dobu hlavního výpočtu.

Mimo metody je každá třída definovaná parametry, s kterými je vytvářena. Pro rozhraní jsou vstupními parametry komplexní indexy lomu jednotlivých prostředí na jeho stranách. Vrstva je definována tloušťkou, komplexním indexem lomu a vlnovou délkou procházející vlny. Difuze je definována funkcí úhlového rozložení, úhlovým vektorem, vlnovou délkou a indexem lomu.

#### 4.4.5 Řešení rozptylové matice

Výsledkem součinu jednotlivých matic rozhraní, vrstev a rozptylu je rozptylová matice  $\mathbf{S}$ . Ta udává vztah mezi vlnami procházejícími strukturou v obou směrech tak, že pro výpočet s počtem prvků v úhlovém vektoru  $n$  můžeme psát:

$$\begin{pmatrix} \varepsilon_{k1a}^{p+} \\ \varepsilon_{k1a}^{p-} \\ \varepsilon_{k1a}^{s+} \\ \varepsilon_{k1a}^{s-} \\ \vdots \\ \varepsilon_{kna}^{p+} \\ \varepsilon_{kna}^{p-} \\ \varepsilon_{kna}^{s+} \\ \varepsilon_{kna}^{s-} \\ \vdots \\ \varepsilon_{u1a}^{p+} \\ \varepsilon_{u1a}^{p-} \\ \varepsilon_{u1a}^{s+} \\ \varepsilon_{u1a}^{s-} \\ \vdots \\ \varepsilon_{una}^{p+} \\ \varepsilon_{una}^{p-} \\ \varepsilon_{una}^{s+} \\ \varepsilon_{una}^{s-} \end{pmatrix} = \begin{pmatrix} S_{11} & \cdots & S_{1n} \\ \vdots & \ddots & \vdots \\ S_{n1} & \cdots & S_{nn} \end{pmatrix} \begin{pmatrix} \varepsilon_{k1b}^{p+} \\ \varepsilon_{k1b}^{p-} \\ \varepsilon_{k1b}^{s+} \\ \varepsilon_{k1b}^{s-} \\ \vdots \\ \varepsilon_{knb}^{p+} \\ \varepsilon_{knb}^{p-} \\ \varepsilon_{knb}^{s+} \\ \varepsilon_{knb}^{s-} \\ \vdots \\ \varepsilon_{u1b}^{p+} \\ \varepsilon_{u1b}^{p-} \\ \varepsilon_{u1b}^{s+} \\ \varepsilon_{u1b}^{s-} \\ \vdots \\ \varepsilon_{unb}^{p+} \\ \varepsilon_{unb}^{p-} \\ \varepsilon_{unb}^{s+} \\ \varepsilon_{unb}^{s-} \end{pmatrix} \quad (4.18)$$

kde horní index  $s$  značí S-polarizaci vlny,  $p$  P-polarizaci vlny,  $+$  směr šíření vlny ve směru dopadající vlny a  $-$  ve směru protichůdném. Spodní index  $k$  značí příslušnost vlny k řešení světla dopadajícího na strukturu,  $u$  světla rozptýleného,  $n$  prvek úhlového vektoru,  $a$  uvádí příslušnost k vlnovému vektoru pravé strany a  $b$  k vektoru levé strany.

V kapitole 2.6 byly odvozeny vztahy pro výpočet celkových Fresnelových koeficientů struktury pro jeden úhel a polarizaci (2.74, 2.75). Důležitým předpokladem byla skutečnost, že vlna dopadá pouze v jednom směru na danou strukturu. Tento předpoklad souvisí s tím, že samostatnou strukturu jako celek nelze osvětlit dvěma vzájemně koherentními svazky světla. Osvětlení struktury zleva i zprava jsou pak řešeny nezávisle a můžeme tedy dostat dva různé případy. Pro pochopení důležitosti tohoto předpokladu lze vycházet z výsledné rovnice struktury pro tento případ:

$$\begin{pmatrix} \varepsilon_a^+ \\ \varepsilon_a^- \end{pmatrix} = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix} \begin{pmatrix} \varepsilon_b^+ \\ \varepsilon_b^- \end{pmatrix} \quad (4.19)$$

Pokud tuto rovnici rozepíšeme do formy rovnic, dostáváme soustavu dvou rovnic:

$$\varepsilon_a^+ = S_{11}\varepsilon_b^+ + S_{12}\varepsilon_b^- \quad (4.20)$$

$$\varepsilon_a^- = S_{21}\varepsilon_b^+ + S_{22}\varepsilon_b^- \quad (4.21)$$

Nyní si vybereme jeden ze směrů, například osvětlení zleva, tedy položíme amplitudu  $\varepsilon_b^-$  rovnou nule. Jelikož nás zajímají pouze koeficienty transmise a reflexe, neboli pouze poměry amplitud, můžeme za dopadající amplitudu  $\varepsilon_a^+$  dosadit jednotku. Výsledkem je soustava dvou rovnic o dvou neznámých. Obdobná situace nastává i v maticové soustavě rovnice 4.18. Dle rovnice 4.17 dostáváme pro  $n_p$  počítaných úhlů soustavu  $8n_p$  matic o  $16n_p$  neznámých. Pokud opět definujeme, že nepolarizovaná vlna dopadá na strukturu pouze ve směru + pod jedním úhlem (pro příklad volíme první prvek maticového vektoru), a pokud nás zajímají jenom poměry, dosadíme do tohoto prvku jednotku, která se v reprezentaci nepolarizované vlny rozloží na dva příspěvky  $1/\sqrt{2}$  pro S a P polarizaci. Rovnice 4.18 dostává podobu:

$$\begin{pmatrix} 1/\sqrt{2} \\ \varepsilon_{k1a}^{p-} \\ 1/\sqrt{2} \\ \varepsilon_{k1a}^{s-} \\ \vdots \\ 0 \\ \varepsilon_{kna}^{p-} \\ 0 \\ \varepsilon_{kna}^{s-} \\ 0 \\ \varepsilon_{u1a}^{p-} \\ 0 \\ \varepsilon_{u1a}^{s-} \\ \vdots \\ 0 \\ \varepsilon_{una}^{p-} \\ 0 \\ \varepsilon_{una}^{s-} \end{pmatrix} = \begin{pmatrix} S_{11} & \dots & S_{1n} \\ \vdots & \ddots & \vdots \\ S_{n1} & \dots & S_{nn} \end{pmatrix} \begin{pmatrix} \varepsilon_{k1b}^{p+} \\ 0 \\ \varepsilon_{k1b}^{s+} \\ 0 \\ \vdots \\ \varepsilon_{knb}^{p+} \\ 0 \\ \varepsilon_{knb}^{s+} \\ 0 \\ \varepsilon_{u1b}^{p+} \\ 0 \\ \varepsilon_{u1b}^{s+} \\ 0 \\ \vdots \\ \varepsilon_{unb}^{p+} \\ 0 \\ \varepsilon_{unb}^{s+} \\ 0 \end{pmatrix} \quad (4.22)$$

Rozeepsáním tohoto maticového součinu bychom již dostali soustavu  $8n_p$  rovnic o  $8n_p$  neznámých. Její řešení ale není snadné jako v předchozím případě. Jelikož díky rozptylu ve struktuře vznikají v matici nediagonální členy, je nutné tuto soustavu řešit strojově. Z knihovny NumPy lze na řešení soustav lineárních rovnic použít funkci `linalg.solve()`. Pro ilustraci mějme soustavu dvou rovnic:

$$3x_1 + 3x_2 = 9 \quad (4.23)$$

$$x_1 + 2x_2 = 8 \quad (4.24)$$

Pro její řešení by funkce `linalg.solve()` měla následující tvar:

```
a = np.array([[3,1], [1,2]])
b = np.array([9,8])
x = np.linalg.solve(a, b)
print(x)

>>>outputs('array([ 2.,  3.])')
```

#### Ukázka kódu 4.3 Řešení soustavy lineárních rovnic v NumPy

Z její syntaxe je očividné (Ukázka kódu 4.3), že funkce očekává dva parametry. Prvním parametrem je matice systému  $A$  a druhým vektor pravé strany rovnic  $b$  [17]. Obecně lze systém lineárních rovnic zapsat:

$$Ax = b \quad (4.25)$$

kde:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}, b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}, x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \quad (4.26)$$

Pokud v rovnici 4.22 označíme neznámé vlny vektorů jednotlivých stran jako neznámé soustavy lineárních rovnic, dostáváme tvar:

$$\begin{pmatrix} b_1 \\ x_1 \\ \vdots \\ b_p \\ x_p \end{pmatrix} = \begin{pmatrix} S_{11} & \cdots & S_{1q} \\ \vdots & \ddots & \vdots \\ S_{q1} & \cdots & S_{qq} \end{pmatrix} \begin{pmatrix} x_{(p+1)} \\ b_{(p+1)} \\ \vdots \\ x_q \\ b_q \end{pmatrix} \quad (4.27)$$

kde  $q$  je počet rovnic a  $p = \frac{q}{2}$ . Při vyjádření součinu matic ve formě lineárních rovnic dostáváme následující:

$$\begin{aligned} b_1 &= S_{11}x_{(p+1)} + S_{12}b_{(p+1)} \dots S_{1(q-1)}x_q + S_{1q}b_q \\ x_1 &= S_{21}x_{(p+1)} + S_{22}b_{(p+1)} \dots S_{2(q-1)}x_q + S_{2q}b_q \\ &\vdots \end{aligned} \quad (4.28)$$

$$b_p = S_{(q-1)1}x_{(p+1)} + S_{(q-1)2}b_{(p+1)} \dots S_{(q-1)(q-1)}x_q + S_{(q-1)q}b_q$$

$$x_p = S_{q1}x_{(p+1)} + S_{q2}b_{(p+1)} \dots S_{q(q-1)}x_q + S_{qq}b_q$$

Pro vytvoření parametrů  $A$ ,  $x$  a  $b$  je vhodné tyto rovnice převést do tvaru, kdy levá strana obsahuje pouze prvky s koeficienty  $b$  a pravá neznámé  $x$ :

$$\begin{aligned}
b_1 - S_{12}b_{(p+1)} \dots - S_{1q}b_q &= S_{11}x_{(p+1)} \dots S_{1(q-1)}x_q \\
0 - S_{22}b_{(p+1)} \dots - S_{2q}b_q &= S_{21}x_{(p+1)} \dots S_{2(q-1)}x_q - x_1 \\
&\vdots
\end{aligned} \tag{4.29}$$

$$\begin{aligned}
b_p - S_{(q-1)2}b_{(p+1)} \dots - S_{(q-1)q}b_q &= S_{(q-1)1}x_{(p+1)} \dots S_{(q-1)(q-1)}x_q \\
0 - S_{q2}b_{(p+1)} \dots - S_{qq}b_q &= S_{q1}x_{(p+1)} \dots S_{q(q-1)}x_q - x_1
\end{aligned}$$

Pro systémovou matici  $\mathbf{A}$  můžeme psát:

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & \dots & 0 & S_{11} & \dots & S_{1(q-1)} \\ -1 & 0 & \dots & 0 & S_{21} & \dots & S_{2(q-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & S_{(q-1)1} & \dots & S_{(q-1)(q-1)} \\ 0 & 0 & \dots & -1 & S_{q1} & \dots & S_{q(q-1)} \end{pmatrix} \tag{4.30}$$

Pro vektor pravé strany  $b$  můžeme psát:

$$b = \begin{pmatrix} b_1 - \sum_{i=1}^{\frac{q}{2}} S_{1(2i)}b_{(p+i)} \\ 0 - \sum_{i=1}^{\frac{q}{2}} S_{2(2i)}b_{(p+i)} \\ \vdots \\ b_l - \sum_{i=1}^{\frac{q}{2}} S_{(q-1)(2i)}b_{(p+i)} \\ 0 - \sum_{i=1}^{\frac{q}{2}} S_{q(2i)}b_{(p+i)} \end{pmatrix} \tag{4.31}$$

V rámci programu se o vytvoření matice koeficientů neznámých a vektoru pravé strany stará funkce `matrixSolver()` (Ukázka kódu 4.4). Ta očekává čtyři parametry: vektor pravé strany rozptylové matice, rozptylová matice, vektor levé strany rozptylové matice a dimenze matice. Funkce nejprve vytvoří ze vstupních parametrů prvky  $\mathbf{A}$  a  $b$  a poté tuto soustavu (4.25) vyřeší. Výstupem funkce jsou vektory jednotlivých stran rozptylové matice s dopočtenými neznámými vlnami.

```

def matrixSolver(left,S,right,dim):
    #Matrix A,b creation
    a = matlib.zeros((dim, dim), dtype=complex, order="F")
    b = matlib.zeros((dim, 1), dtype=complex, order="F")

    k = 0
    #Matrix A,b elements calculation
    for i in range(0,len(left)): ...

    x = np.linalg.solve(a, b)

    k = 0
    #Wave vector update
    for i in range(0,len(left)): ...

    return(left,right)

```

*Ukázka kódu 4.4 Funkce pro řešení vlnových vektorů*



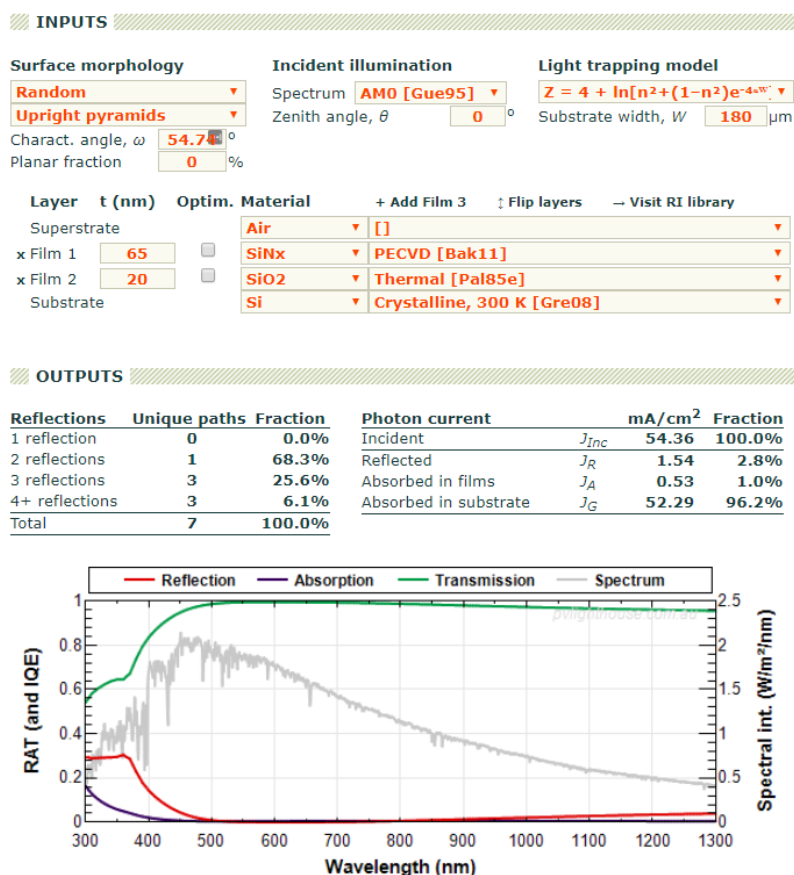
## 5 Porovnání s existujícími programy

V oblasti rozptylu světla existuje řada komerčně dostupných programů. Ve většině případů jsou ale zaměřené na popis chování světla při simulacích osvětlení scén v počítačové grafice. Z open source zdrojů jsou v této oblasti nejrozšířenějšími variace Python knihovny PyMieScatt, Blender Scattering model a C++ knihovna SCATMECH.

V oblasti simulací interakce světla s rovinnou strukturou, kterou se vyvíjený program zabývá, je dostupných programů méně. V této kapitole bude vyvinutý program RLayer podrobněji porovnán s webovou aplikací PV Lighthouse a Matlab programem OPTOS vyvíjeným institutem Fraunhofer ISE.

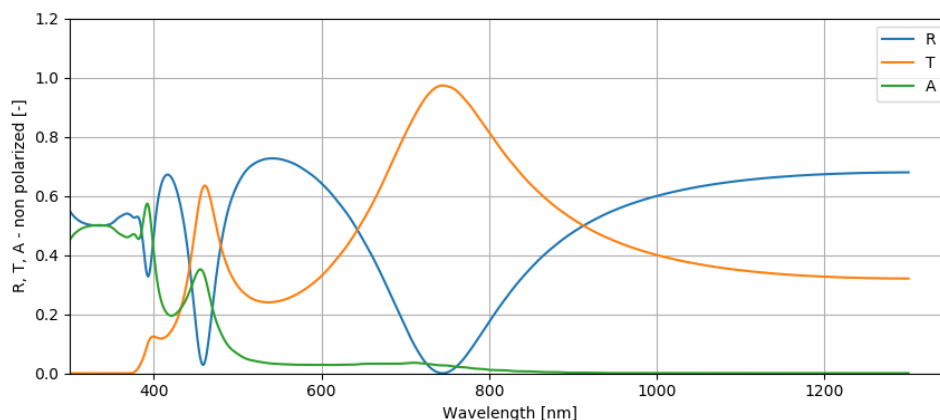
### 5.1 PV Lighthouse

PV Lighthouse je volně dostupná webová aplikace vyvíjená jako kolektivní dílo řady autorů. Hlavním zaměřením této stránky je simulace operací spojených se studiem solárních článků. Pro optické simulace je aktuálně dostupný výpočetní modul OPAL 2 [18]. Program umožňuje návrh struktury, morfologie povrchu a dopadajícího spektra. Pro zadaný úhel dopadu poté počítá reflexi, absorpci a transmisi (Obrázek 5.1). Materiály vrstev jdou zvolit z výchozí databáze nebo je lze uživatelem nahrát.

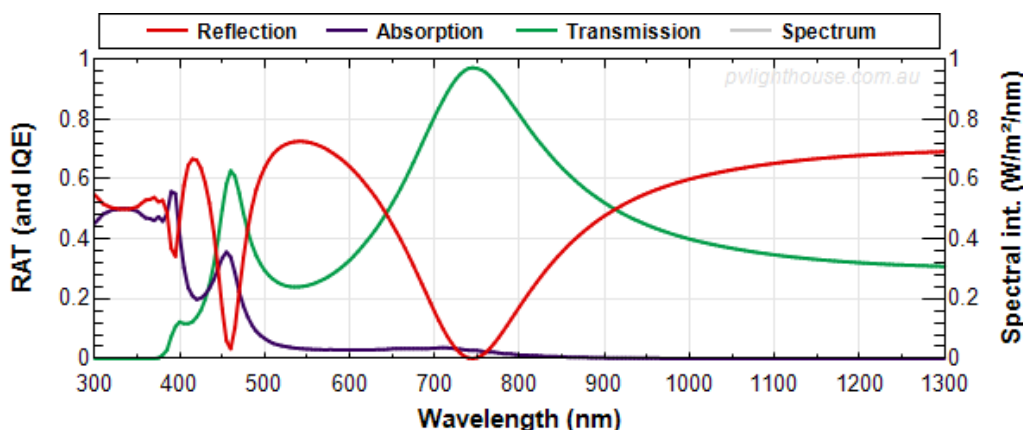


Obrázek 5.1 Webový program OPAL 2

Použitý výpočetní model je založený na metodě Ray-tracing. Oproti maticovému výpočtu použitému v práci OPAL 2 hledá pro každý dopadající paprsek všechny možné cesty materiálem a z nich počítá jednotlivé koeficienty.



Obrázek 5.2 Simulace v programu R Lay



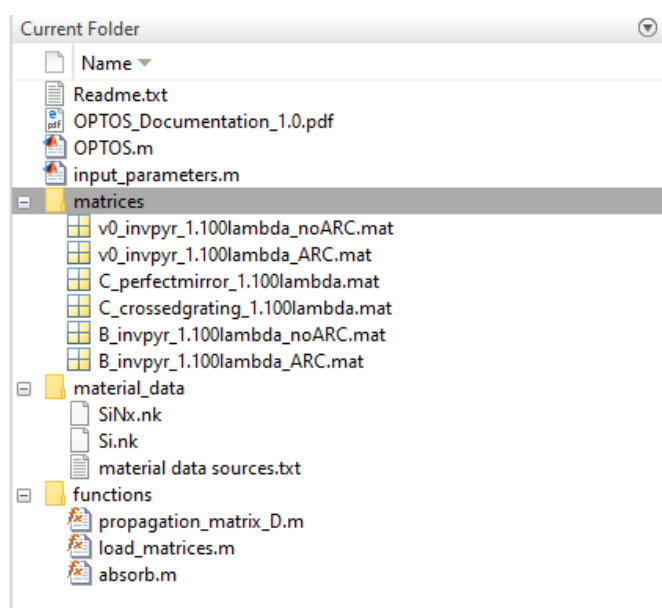
Obrázek 5.3 Simulace v programu OPAL 2 [18]

Program OPAL 2 neumožňuje vizualizaci struktury a difuzi ve struktuře. Pro porovnání výsledků byla zvolena struktura umístěná ve vzduchu složená z  $\text{SiO}_2$  [11] vrstvy o tloušťce 20 nm a Si [7] vrstvy o tloušťce 100 nm. Do obou programů byla nahrána identická materiálová data a rozsah simulace byl stanoven od 300 nm do 1300 nm. Jelikož OPAL 2 (Obrázek 5.3) neumožňuje dělení na jednotlivé polarizace, byl pro porovnání v programu R Lay (Obrázek 5.2) zvolen výpočet reflexe, transmise a absorpce pro nepolarizované světlo. Výsledné průběhy jsou identické.

## 5.2 Fraunhofer ISE OPTOS

Fraunhofer OPTOS je metoda výpočtu vyvíjená institutem Fraunhofer ISO. Její primární zaměření je optické modelování texturovaných povrchů pomocí maticového formalismu. Základní podoba je distribuována pod GNU licenci, aplikace pro konkrétní případy je poskytována jako komerční produkt.

Program je dostupný jako souhrn zdrojových souborů programu Matlab. Ve volně dostupné variantě obsahuje program sadu předem vytvořených matic pro určitou vlnovou délku a typ povrchu (Obrázek 5.4). Samotný výpočet není bez manuální modifikace matic možný. Program neobsahuje grafické uživatelské rozhraní a je dodáván s databází dvou materiálů. Metoda OPTOS je založena na funkčním programování s minimální možností škálování programu.



Obrázek 5.4 Komponenty programu OPTOS

OPTOS tedy není celistvě dodávané programové řešení. Jedná se spíše o souhrn funkcí vyvinuté metodiky, na kterém lze stavět vlastní řešení. Z matematického hlediska sdílí s programem RLayer diskretizaci úhlů, pro které je výpočet prováděn, a maticový přístup k řešení.



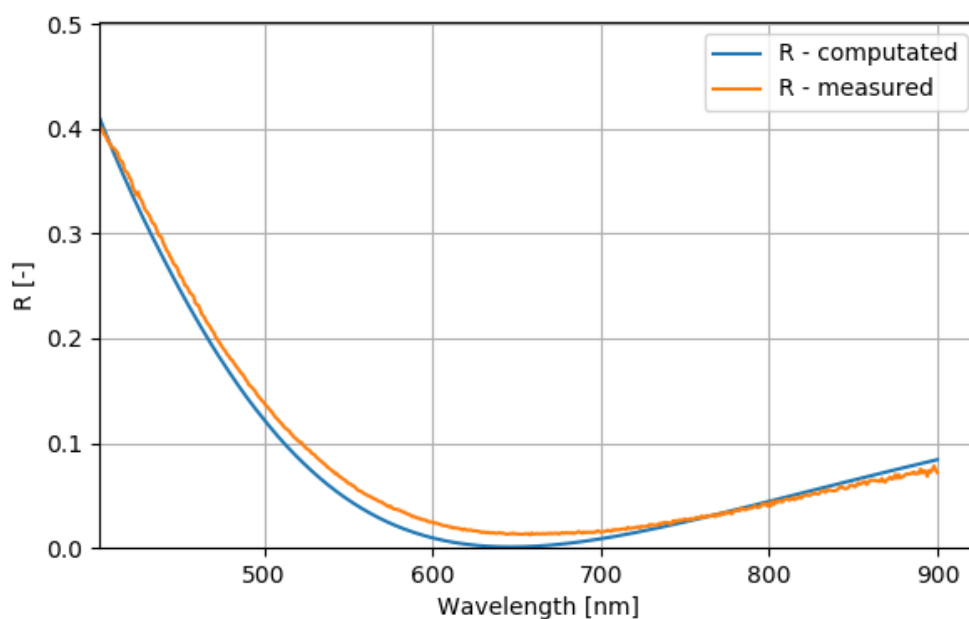
## 6 Porovnání s experimentálními daty

Pro porovnání s experimentálními daty byla použita data naměřená v rámci bakalářské práce *Studium antireflexních vrstev barevných solárních článků pomocí vláknového spektrometru* realizované na ČVUT FEL katedra elektrotechnologie [19]. Porovnání naměřených dat reflexe s výstupem simulací je provedeno pro dva vzorky solárních článků. Strukturou se jedná o antireflexní vrstvu  $\text{SiN}_x$  umístěnou na substrátu z krystalického křemíku. Podrobné parametry vzorků (Tabulka 6.1) jsou:

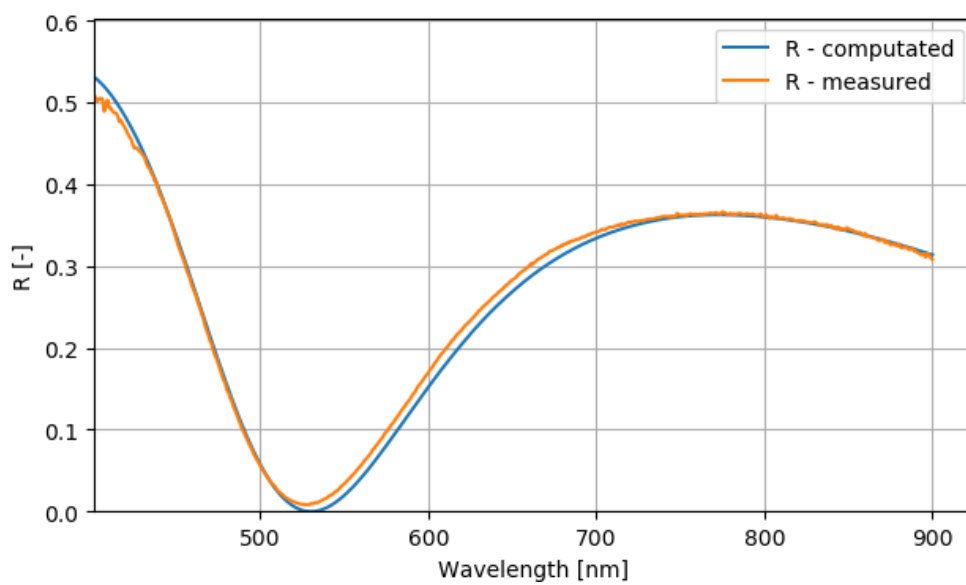
*Tabulka 6.1* Vzorky solárních článků

vzorek	substrát	AR vrstva	tloušťka AR vrstvy [nm]
FF04	c-Si	$\text{SiN}_x$	80
FF09	c-Si	$\text{SiN}_x$	195

Při simulaci je struktura navržena jako samostatná vrstva materiálu  $\text{SiN}_x$  [13] umístěná mezi prostředím vzduchu a krystalického křemíku [7]. Simulace probíhá v rozsahu 400 – 900 nm pro úhel dopadu  $0^\circ$ . Výsledky simulací jsou vyneseny do společného grafu samostatně pro vzorek FF04 (Obrázek 6.1) a FF09 (Obrázek 6.2).



*Obrázek 6.1* Porovnání simulace a dat pro vzorek FF04



*Obrázek 6.2 Porovnání simulace a dat pro vzorek FF09*

## 7 Závěr

Cílem diplomové práce bylo popsání matematického modelu rozptylu světla ve struktuře pomocí koherentního maticového formalismu. V teoretické části byly odvozeny jednotlivé vztahy nutné pro konstrukci rozptylové matice svazující světelné vlny na jednotlivých stranách simulované struktury. V teoretické části byl popsán zvolený programovací jazyk Python a důvody, pro volbu tohoto jazyka, zejména nároky na efektivitu maticových výpočtů. Dále jsou popsány základní principy fungování jednotlivých knihoven využitých pro realizaci simulačního programu.

V praktické části byl vyvinut simulační program struktur homogenních rovinných vrstev RLayer. Jeho základní funkcionalitou je schopnost vytvoření jednotlivých matic charakterizujících rozhraní a vrstvy definované struktury. Parametry struktury jsou uživatelsky zadávány přes grafické uživatelské rozhraní (GUI). Uživatel může navrhnout libovolnou strukturu složenou z materiálů obsažených ve výchozí databázi programu, nebo může nahrát parametry vlastních materiálů z externího souboru. Výsledná podoba struktury je v programu vizualizována. Pro zadaný rozsah vlnových délek a úhel dopadu poté RLayer počítá reflexi, transmisi a absorpci světla pro obě polarizace. Závislosti jsou uživateli zobrazeny v grafu s možností manipulace (přiblížení, ořez). Výstup je možné exportovat do standardních grafických a textových formátů.

Další důležitou funkcionalitou je možnost umístění difuzních rovin do struktury. Návrh umístění probíhá přes GUI a roviny jsou zobrazeny ve vizualizaci. Pro výpočet difundovaného světla je v praktické části odvozena diskretizace úhlového vektoru, pro který výpočet probíhá. Složitost maticového výpočtu roste lineárně s množstvím úhlů, pro které je propagace světla strukturou počítána. Tato složitost je uživatelsky nastavitelná. Pro tímto způsobem vytvořenou rozměrnou rozptylovou matici charakterizující strukturu je odvozen způsob převedení transformační rovnice vlnových vektorů a rozptylové matice na soustavu lineárních rovnic. Soustava je poté numericky řešena.

V závěru práce je program RLayer porovnán s dostupnými programovými řešeními podobného zaměření PV Lighthouse OPAL 2 a Fraunhofer ISE OPTOS. Výstup simulace je rovněž porovnán s experimentálními daty naměřenými v rámci bakalářské práce realizované v roce 2016 na katedře elektrotechnologie ČVUT FEL [19]. Simulace tvarově následuje experimentální data v celém rozsahu.





## Použité zdroje

- [1] GRAND VIEW RESEARCH. Optical Coating Market Analysis. 978-1-68038-740-7. 2016, 145.
- [2] SERWAY, Raymond A., John W. JEWETT a Vahé PEROOMIAN. *Physics for scientists and engineers with modern physics*. Ninth edition. Boston, MA: Brooks/Cole, Cengage Learning, 2014. ISBN 978-1-133-95405-7.
- [3] DOI, Takuma, Kouji TOYODA a Yoshihisa TANIMURA. Effects of phase changes on reflection and their wavelength dependence in optical profilometry. *Applied Optics* [online]. 1997, **36**(28), 7157. ISSN 0003-6935, 1539-4522. Dostupné z: doi:10.1364/AO.36.007157
- [4] DRIGGERS, Ronald G., ed. *Encyclopedia of optical engineering*. New York: Dekker, 2003. ISBN 978-0-8247-0940-2.
- [5] ŠTEFAN VIŠŇOVSKÝ. Optika tenkých vrstev a vrstevnatých struktur. In: . FU UK.
- [6] PYTHON SOFTWARE FOUNDATION. *Python Logo* [online]. Dostupné z: <https://www.python.org/community/logos/>
- [7] GREEN, Martin A. a Mark J. KEEVERS. Optical properties of intrinsic silicon at 300 K. *Progress in Photovoltaics: Research and Applications* [online]. 1995, **3**(3), 189–192. ISSN 10627995, 1099159X. Dostupné z: doi:10.1002/pip.4670030303
- [8] TREHARNE, R. E., A. SEYMOUR-PIERCE, K. DUROSE, K. HUTCHINGS, S. RONCALLO a D. LANE. Optical Design and Fabrication of Fully Sputtered CdTe/CdS Solar Cells. *Journal of Physics: Conference Series*. 2011, **286**(1), 012038.
- [9] RAKIĆ, Aleksandar D. Algorithm for the determination of intrinsic optical constants of metal films: application to aluminum. *Applied Optics* [online]. 1995, **34**(22), 4755. ISSN 0003-6935, 1539-4522. Dostupné z: doi:10.1364/AO.34.004755
- [10] QUERRY, M R. *Optical Constants*. ADA158623. B.m.: MISSOURI UNIV-KANSAS CITY. 1985.
- [11] GAO, Lihong, Fabien LEMARCHAND a Michel LEQUIME. Exploitation of multiple incidences spectrometric measurements for thin film reverse engineering. *Optics Express* [online]. 2012, **20**(14), 15734. ISSN 1094-4087. Dostupné z: doi:10.1364/OE.20.015734
- [12] STELLING, Christian, Chetan R. SINGH, Matthias KARG, Tobias A. F. KÖNIG, Mukundan THELAKKAT a Markus RETSCH. Plasmonic nanomeshes: their ambivalent role as transparent electrodes in organic solar cells. *Scientific Reports* [online]. 2017, **7**, 42530. ISSN 2045-2322. Dostupné z: doi:10.1038/srep42530
- [13] Refractive index of Silicon Nitride. *Filmetrics* [online]. 20. březen 2016. Dostupné z: <https://www.filmetrics.com/refractive-index-database/Si3N4/Silicon-Nitride-SiN>

- [14] E JONES, E OLIPHANT a P PETERSON. *SciPy: Open Source Scientific Tools for Python* [online]. 2001. Dostupné z: <http://www.scipy.org>
- [15] BURNETT, John H., Simon G. KAPLAN, Eric STOVER a Adam PHENIS. Refractive index measurements of Ge. In: Paul D. LEVAN, Ashok K. SOOD, Priyalal WIJEWARNASURIYA a Arvind I. D'SOUZA, ed. [online]. 2016, s. 99740X [vid. 2018-04-22]. Dostupné z: doi:10.1117/12.2237978
- [16] JAEGER, Klaus, Rene van SWAAIJ a Miro ZEMAN. The Scalar Scattering Theory: A Multi-Functional Tool for Optimizing Scattering in Thin-Film Silicon Solar Cells. In: *Renewable Energy and the Environment Optics and Photonics Congress* [online]. B.m.: Optical Society of America, 2012, s. PT3C.7. Dostupné z: doi:10.1364/PV.2012.PT3C.7
- [17] OLIPHANT, Travis E. *Guide to NumPy*. Austin, Tex.: Continuum Press, 2015. ISBN 978-1-5173-0007-4.
- [18] K.R. MCINTOSH a S.C. BAKER-FINCH. OPAL 2: Rapid optical simulation of silicon solar cells. In: *Proceedings of the 38th IEEE Photovoltaic Specialists Conference*. 2012.
- [19] RADEK NEVYHOŠTĚNÝ. *Studium antireflexních vrstev barevných solárních článků pomocí vláknového spektrometru*. Praha, 2016. Bakalářská práce. České vysoké učení technické v Praze.

## Seznam obrázků

<b>Obrázek 2.1</b> Elektrické pole na rozhraní .....	18
<b>Obrázek 2.2</b> Magnetické pole na rozhraní .....	19
<b>Obrázek 2.3</b> P-polarizovaná vlna na rozhraní.....	20
<b>Obrázek 2.4</b> S-polarizovaná vlna na rozhraní .....	22
<b>Obrázek 2.5</b> Brewsterův úhel na rozhraní vzduch-křemík .....	23
<b>Obrázek 2.6</b> Jednovrstvá struktura .....	24
<b>Obrázek 2.7</b> Izotropní mnohovrstvá struktura .....	26
<b>Obrázek 3.1</b> Logo jazyka Python [6] .....	33
<b>Obrázek 3.2</b> Paměťové rozložení C a Fortran .....	36
<b>Obrázek 3.3</b> Ilustrační graf Matplotlib .....	38
<b>Obrázek 4.1</b> Struktura programu RLayer .....	39
<b>Obrázek 4.2</b> Okno Computation programu RLayer .....	40
<b>Obrázek 4.3</b> Příklad výstupního grafu programu RLayer .....	41
<b>Obrázek 4.4</b> Okno Structure programu RLayer.....	42
<b>Obrázek 4.5</b> Okno Scatter programu RLayer .....	43
<b>Obrázek 4.6</b> Interpolace zdrojových dat .....	44
<b>Obrázek 4.7</b> Rozložení sinu úhlu v úhlovém vektoru pro 13 prvků.....	47
<b>Obrázek 4.8</b> Difuze vlny na rozhraní.....	48
<b>Obrázek 4.9</b> Difuze vlny ve vrstvě.....	49
<b>Obrázek 5.1</b> Webový program OPAL 2.....	57
<b>Obrázek 5.2</b> Simulace v programu RLayer .....	58
<b>Obrázek 5.3</b> Simulace v programu OPAL 2 [18] .....	58
<b>Obrázek 5.4</b> Komponenty programu OPTOS.....	59
<b>Obrázek 6.1</b> Porovnání simulace a dat pro vzorek FF04 .....	61
<b>Obrázek 6.2</b> Porovnání simulace a dat pro vzorek FF09 .....	62

## Seznam ukázek kódu

<b>Ukázka kódu 3.1</b> Duck-typing v jazyce Python.....	34
<b>Ukázka kódu 3.2</b> Měření výpočetní doby operací na NumPy poli.....	37
<b>Ukázka kódu 4.1</b> Interpolace vstupních dat.....	44
<b>Ukázka kódu 4.2</b> Třída Layer výpočetní funkce.....	50
<b>Ukázka kódu 4.3</b> Řešení soustavy lineárních rovnic v NumPy.....	54
<b>Ukázka kódu 4.4</b> Funkce pro řešení vlnových vektorů .....	56

## Seznam tabulek

<b>Tabulka 3.1</b> Základní číselné typy .....	35
<b>Tabulka 4.1</b> Zabudované materiály programu RLayer .....	42
<b>Tabulka 4.2</b> Parametr data maticových objektů.....	51
<b>Tabulka 6.1</b> Vzorčky solárních článků .....	61