



PŘÍRODOVĚDECKÁ FAKULTA
UNIVERZITA PALACKÉHO V OLMOUCI

KATEDRA
EXPERIMENTÁLNÍ FYZIKY

Bakalářská práce

Vizualizace optických zobrazení
v programu
Wolfram Mathematica

Autor	Rostislav Danel
Vedoucí práce	Mgr. Jan Říha, Ph.D.
Studijní program	B1701 Fyzika
Studijní obor	1701R003 Fyzika se zaměřením na vzdělávání
Forma studia	Prezenční
Rok	2019

Jméno a příjmení autora	Rostislav Danel
Název práce	Vizualizace optických zobrazení v programu Wolfram Mathematica
Typ práce	Bakalářská
Pracoviště	Katedra experimentální fyziky
Vedoucí práce	Mgr. Jan Říha, Ph.D.
Rok obhajoby práce	2019
Počet stran	48
Počet příloh	1
Jazyk	český
Abstrakt	Zabývali jsme se vizualizací optických zobrazení v paraxiálním prostoru. Vytvořili jsme knihovnu funkcí v prostředí Wolfram Mathematica pro stopování paprsků optickou soustavou složenou z lámavých ploch, následně jsme tuto knihovnu aplikovali při tvorbě dynamických modelů a řešení klasických příkladů z paprskové optiky, které jsou počítány v úvodních kurzech studia optiky na Přírodovědecké fakultě Univerzity Palackého. Vizualizované soustavy mohou být použity i pro vysvětlení principů paprskové optiky na středních školách. V bakalářské práci je knihovna okomentována a může být použita pro tvorbu vlastních vizualizací a řešení příkladů.
Klíčová slova	Wolfram Mathematica, paprsková optika, dynamická vizualizace, paraxiální paprsky

Author's first name and surname	Rostislav Danel
Title	Visualization of Optical Imaging in Software Wolfram Mathematica
Type of thesis	Bachelor
Department	Department of Experimental Physics
Supervisor	Mgr. Jan Říha, Ph.D.
Year of presentation	2019
Number of pages	48
Number of appendices	1
Language	czech
Abstract	We were concerning the issue of optical imaging in paraxial space. We created library of functions in software Wolfram Mathematica for tracing rays with optical system composed of refracting surfaces. Then we used the library for creating dynamic models and solving regular problems of geometric optics which are solved in introductory lectures of optics at Faculty of Science of Palacky University. Visualizations of optical systems can be used for explanation of geometric optics basic principles at high schools. The library is well commented and it can be used for creating visualizations and solving problems.
Keywords	Wolfram Mathematica, geometric optics, dynamic visualization, paraxial rays

PROHLÁŠENÍ

*Prohlašuji, že jsem předloženou bakalářskou práci vypracoval samostatně pod vedením Mgr. Jana Říhy, Ph.D.
a že jsem použil zdrojů, které cituji a uvádím v seznamu použitých zdrojů.*

V Olomouci dne

.....
podpis

Rád bych poděkoval vedoucímu mé práce panu Mgr. Janu Říhovi, Ph.D. za cenné rady a za čas, který mi věnoval. Dále bych chtěl poděkovat Mgr. Davidu Smrčkovi za jeho pomoc při práci s jazykem Wolfram.

Obsah

Úvod	8
1. Úvod do geometrické optiky	9
1.1. Zákon odrazu	9
1.2. Zákon lomu	10
1.3. Zobrazení zrcadly	11
1.3.1. Zobrazovací rovnice sférického zrcadla	12
1.3.2. Zvětšení obrazu sférickým zrcadlem	13
1.4. Zobrazení lámavými plochami	14
1.4.1. Znaménková konvence pro lámavé plochy	14
1.4.2. Chod paprsku jednou kulovou lámavou plochou	14
1.4.3. Stopování paprsku skrze konečné množství sférických lámavých ploch	15
1.4.4. Paprsky v paraxiálním prostoru	16
1.4.5. Zobrazovací rovnice lámavé plochy, příčné zvětšení a ohniska	17
1.4.6. Kardinální body optických soustav a jejich určení, úhlové zvětšení	18
1.4.7. Určení zvětšení optické soustavy a polohy hlavních bodů	19
1.4.8. Osové zvětšení optické soustavy	19
1.4.9. Čočky	19
2. Představení prostředí Wolfram Mathematica	20
2.1. Symbolické výrazy	20
2.2. List	21
2.3. Association List	21
2.4. Logické výrazy	22
2.5. Iterace a podmínky	22
2.6. Lokální proměnné a volitelné parametry funkcí	22
2.7. Matematické funkce	22
2.8. Plot, ContourPlot, Graphics a Show	22
2.9. Manipulate, Control, FormObject, Failure a Dynamic	24
3. Simulace chodu paprsku optickou soustavou	26
3.1. Funkce vytvořené pro trasování paprsků optickou soustavou	26
3.1.1. Repräsentace světelných paprsků	26
3.1.2. Vytvoření lámavé plochy	26
3.1.3. Přidání ohniskových vzdáleností, optické mohutnosti a středu lámavé plochy	27
3.1.4. Souřadnice průniku lámavé plochy a paprsku	27
3.1.5. Výpočet souřadnic obrazů předmětových bodů po zobrazení přes konečný počet lámavých ploch	27
3.1.6. Trasování paprsků rovnoběžných s optickou osou	28
3.1.7. Výpočet osového zvětšení optickou soustavou	29
3.1.8. Hlavní body a ohniska optické soustavy	29
3.2. Vizualizace dosažených výsledků	30
3.2.1. Vykreslování paprsků	30
3.2.2. Vizualizace předmětů a obrazů	30
3.2.3. Určení mezí pro vykreslovanou oblast	31
3.2.4. Grafická reprezentace lámavých ploch a význačných bodů	32
3.3. Tvorba dynamického modelu	32
3.3.1. Kontrola zadání nekonečna	32
3.3.2. Tvorba rozhraní a volba počtu čoček	32
3.4. Návod k použití dynamického rozhraní	33
3.4.1. Parametry předmětů a paprsků	33
3.4.2. Parametry optické soustavy	34
3.4.3. Vlastnosti optické soustavy	34
3.4.4. Vlastnosti předmětů a obrazů	34
3.5. Ukázky grafických výstupů	36
3.6. Optická lavice	40

4. Příklady řešené pomocí simulace a přímým výpočtem	41
4.1. 1. příklad	41
4.2. 2. příklad	43
4.3. 3. příklad	44
Závěr	46
Seznam použitých zdrojů	47
Příloha 1. Kód na CD	48

Úvod

Stephen Wolfram píše o jazyce Wolfram:

“It’s designed to make it as easy as possible to describe what you want, making use of huge amounts of knowledge that are built into the language”.^[1]

Jazyk Wolfram považuje vše za výraz, a díky této architektuře je pro uživatele použití tohoto jazyka snadné. Může tak rychle přejít k realizaci jeho projektů. Obzvláště vhodný je pro symbolické matematické výpočty a práci s databází dat obsažených v jazyce.

Prostředí Wolfram Mathematica umožňuje jednoduchým způsobem dynamicky vizualizovat množství dějů, které vidíme ve světě kolem nás. A právě této vlastnosti softwaru Mathematica jsme využili při konstrukci aplikace znázorňující šíření světelného paprsku optickou soustavou.

Cílem předložené bakalářské práce je vytvoření aplikace v prostředí Mathematica pro trasování paraxiálního paprsku optickou soustavou složenou ze sférických lámavých ploch a vypracování přehledu funkcí, kterých bylo k naprogramování modelu potřeba. Vytvoření modelu musí zákonitě předcházet studium jazyka Wolfram a fyzikální podstaty děje. Součástí práce je i řešení složitějších optických soustav a vizualizace středoškolských úloh. Tento model by měl být použitelný při výuce úvodních kurzů optiky na Přírodovědecké fakultě Univerzity Palackého. Naším cílem je i vytvoření knihovny funkcí,¹ která by byla využitelná při tvorbě vlastních modelů z paprskové optiky nebo při řešení příkladů.

Motivací pro výběr tématu *Vizualizace optických zobrazení v programu Wolfram Mathematica* byla potřeba podílet se na naprosto nezbytném začleňování informatiky do dalších oblastí výuky na středních školách, především přírodních věd a matematiky. Díky jednoduchosti je jazyk Wolfram mimořádně vhodný pro první setkání s programováním a zmíněná databáze v jazyce poskytuje prostor pro neobvyklé a zajímavé příklady, které je v jiných jazycích obtížné vytvořit.

Předložený text je rozčleněn do čtyř kapitol. V kapitole 1 rozebíráme metodu sledování paprsku optickou soustavou v paraxiálním přiblížení a postupy vedoucí k zjištění dalších parametrů optické soustavy. Zabývali jsme se zde odvozením základů geometrické optiky. Začali jsme definicí indexu lomu, odvodili jsme zákon odrazu a zavedli zákon lomu. Vyvodili jsme zobrazení kulovými zrcadly a sférickými lámavými plochami, kterým jsme se věnovali nejvíce. Kapitole jsme zakončili velmi stručným zavedením čoček, jak je chápeme v našem modelu a zmínkou o matematickém formalismu.

Ve 2. kapitole se zabýváme jazykem Wolfram a build-in funkcemi prostředí Mathematica v rozsahu užitém ve vytvořené aplikaci. V kapitole 3 již vytváříme zmíněnou knihovnu i aplikaci a přidáváme ukázky výstupů. Ve 4. kapitole řešíme tři klasické příklady na trasování paprsků optickou soustavou.

¹Z funkcí této knihovny bychom vytvořili i zadanou aplikaci.

1. Úvod do geometrické optiky

Geometrická optika je model šíření světla popisující průchod paprsků světla optickým prostředím, přičemž *paprsek světla* je čirá abstrakce; dále budeme světelný paprsek považovat za přímku ve dvourozměrném nebo trojrozměrném prostoru. Tento model nedokáže správně popsat celou řadu optických jevů, ku příkladu interferenci, či polarizaci světla. Ovšem je vhodný pro stopování průchodu paprsků *optickou soustavou*, tuto znalost pak můžeme použít pro konstrukci základních přístrojů.

“Optická soustava je uspořádání optických prostředí, které mění směr chodu paprsků. Postup, kterým získáme optické obrazy bodů (tzv. předmětů), se nazývá *optické zobrazení*”. [2]

Optickou soustavou budeme v dalším textu rozumět systém složený z lámavých kulových² ploch, které rozdělují dvě prostředí s převážně různou optickou hustotou. Mějme dvě prostředí. Prostor budeme nazývat *optický hustší*, bude-li rychlost světla v tomto prostředí nižší, než v prostředí druhém. Duálně můžeme zavést pojem *optický řidší prostředí*. Mírou hustoty prostředí pak bude *index lomu*, daný vztahem

$$n = \frac{c}{v}, \quad (1)$$

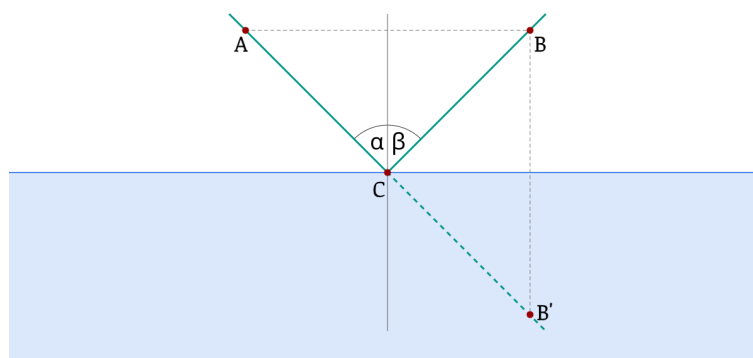
kde c je rychlost šíření světelného paprsku ve vakuu a v je rychlost světelného paprsku v prostředí. Jelikož rychlost šíření paprsku v prostředí nemůže být nikdy vyšší než rychlost šíření světelného paprsku ve vakuu, je index lomu číslo vždy větší nebo rovné jedné.

V dalším textu budeme hojně využívat základních geometrických konstrukcí a zobrazení. Budeme uvažovat vždy dvojrozměrný eukleidovský prostor, v některých případech trojrozměrný. Paprsky budeme považovat vždy za přímky, jelikož nebudeme uvažovat optická prostředí se spojitě se měnícím indexem lomu, ani jiné speciální případy. Zrcadla a lámavé plochy budou představovat přímky a části kružnic ve dvojrozměrném prostoru a roviny a části kulových ploch ve trojrozměrném.

1.1. Zákon odrazu

Dle [3, 4] používal pojem paprsku již HERON ALEXANDRIJSKÝ kolem roku 125 př. n. l. a spojoval jej s principem *nejkratší dráhy světla*, nazývaným též Heronův princip. Z něj a z předpokladu přímočarého šíření paprsků světla odvodil *zákon odrazu*.

Uvažujme body A, B a zrcadlo (obr. 1). Bod A je zdrojem světelných paprsků dopadajících na plochu zrcadla. Paprsky nepronikají skrze zrcadlo, ale mění směr šíření tak, aby zachovaly princip nejkratší dráhy. Chceme-li určit dráhu paprsku, který prochází bodem B a vychází z bodu A, musíme najít bod C, ve kterém se paprsek odráží. Takový bod C nalezneme jako průnik úsečky představující zrcadlo a spojnice bodů B a B', který je obrazem bodu B v osové souměrnosti podle úsečky představující zrcadlo. Touto geometrickou konstrukcí jsme našli nejkratší cestu mezi body A a B takovou, že se paprsek odráží na ploše zrcadla. Samozřejmě existuje kratší cesta a to spojnice bodu A a B, tu ale neuvažujeme.



Obrázek 1 | Znázornění zákona odrazu.

²V následujícím textu užíváme různě synonym kulový a sférický.

Na obrázku 1 jsou vyznačeny úhly α a β , které představují po řadě úhel dopadu a úhel odrazu. Z předchozího tedy vyplývá vztah

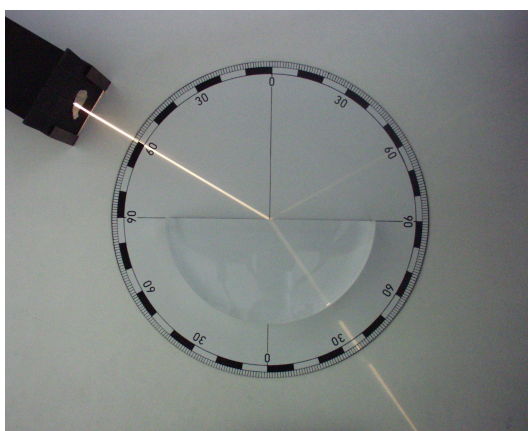
$$\alpha = \beta, \quad (2)$$

nazývaný zákonem odrazu.

Představíme-li si chod paprsku trojrozměrným prostorem, nebude nám rovnost 2 stačit k určení cesty odraženého paprsku. Musíme uvažovat takzvanou *rovinu dopadu*, která je dána přímkou představující dopadající paprsek a kolmým průmětem této přímky do roviny představující zrcadlo. Platí tvrzení: odražený paprsek náleží rovině dopadu.

1.2. Zákon lomu

Snelliův zákon nebo také zákon lomu popisuje chování světelného paprsku na rozhraní dvou prostředí. Dopadne-li paprsek na takové rozhraní, pak se v závislosti na indexu lomu obou prostředí a úhlu dopadu buď odrazí, jak jsme popsali výše (rozhraní se chová jako zrcadlo), nebo projde do druhého prostředí. Často proběhnou oba jevy současně. Situaci, kdy paprsek pronikne do druhého prostředí nazýváme lom. Jak můžeme vidět na obrázku 2, paprsek nepokračuje přímo, ale změní směr šíření.



Obrázek 2 | Znázornění zákona lomu a odrazu. Fotografie převzata od Zátonyi Sándora.³

Změnu šíření paprsku světla na rozhraní vzduchu a vody popsal v roce 140 n. l. KLAUDIUS PTOLEMAIOS. [5] Jím určené hodnoty jsou zaneseny do tabulky 1. Dle HUYGENSOVA svědectví vyslovil zákon lomu z empiricky získaných dat holandský matematik WILLIBRORD SNELL (1591-1626) v roce 1621. [3, 5]

Snellovo pravidlo dle [5] zní:

“Je-li úhel ve vzduchu α_1 a úhel ve vodě α_2 , sinus α_1 je roven nějakému konstantnímu násobku α_2 ”.

Můžeme tedy psát

$$\sin \alpha_1 = n \sin \alpha_2,$$

kde $n = 1.33$ (pro rozhraní voda, vzduch) [5] je poměr indexů lomů dvou prostředí. Pro prostředí s konstantními indexy lomu můžeme psát

$$n_1 \sin \alpha_1 = n_2 \sin \alpha_2, \quad (3)$$

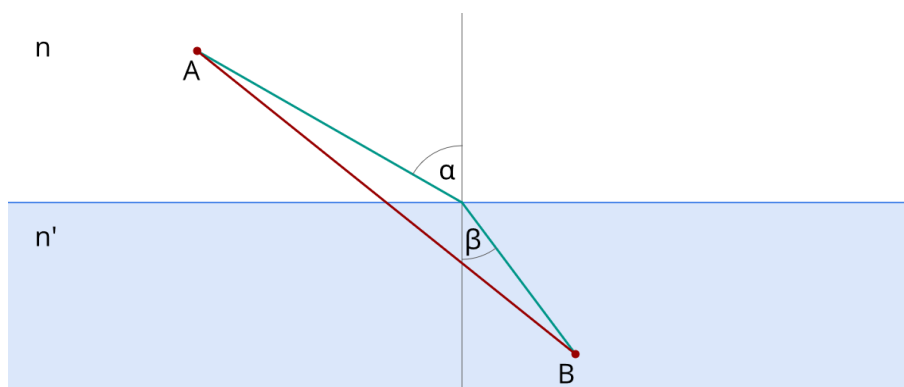
kde n_1 je index lomu prvního prostředí a n_2 je index lomu druhého prostředí. Tento tvar je známý již ze středních škol.

Ve stejném století jako WILLEBRORD SNELL odvodil stejný zákon lomu i matematik PIERRE DE FERMAT (1601-1665), ten upřesnil Heronův princip.[4] Už nemluvil o nejkratší dráze, ale o *nejkratší optické dráze*, tedy “cestě” po které se světlo dostane z bodu A do bodu B za nejkratší čas. Používá se označení princip nejkratšího času šíření nebo Fermatův princip.[4] Rozdíl mezi Heronovým a Fermatovým principem, resp. nejkratší dráhou a nejkratší optickou dráhou, je znázorněn na obrázku 3.

³Autor: Zátonyi Sándor (if.) Fizped (talk) – Vlastní dílo, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=2845439>

Úhel dopadu ve vzduchu	Ptolemaiem určený úhel lomu ve vodě	Úhel lomu dle Snellova zákona ve vodě
10°	8°	7,5°
20°	15,5°	15°
30°	22,5°	22°
40°	29°	29°
50°	35°	35°
60°	40,5°	40,5°
70°	45,5°	45°
80°	50°	48°

Tabulka 1 Úhly lomu na rozhraní vzduchu a vody určené KALAUDIEM PTOLEMAIEM a dle Snellova zákona. Data převzata z [5] (str. 347).



Obrázek 3 Dvě prostředí s různou optickou hustotou. Zeleně je znázorněna dráha dle Fermatova principu a červeně dle Heronova. Zelená dráha neodpovídá skutečnosti, viz srovnání s obrázkem 2.

1.3. Zobrazení zrcadly

“Nejjednodušší a současně i nejstarší optickou soustavou je *rovinné zrcadlo*”. [6]

Schematicky jsou předmět, rovinné zrcadlo a obraz znázorněny na obrázku 4. Obraz zrcadla je *zdánlivý, nezvětšený, ani nezmenšený a zrcadlově převrácený*.

Dále se budeme věnovat *sférickým zrcadlům*. Jedná se o zrcadlovou plochu na povrchu sférické plochy. Zrcadlo může být na vnější, resp. vnitřní, straně kulové plochy, pak jej nazýváme *vypuklé (konvexní)*, resp. *vyduté (konkávní)*.

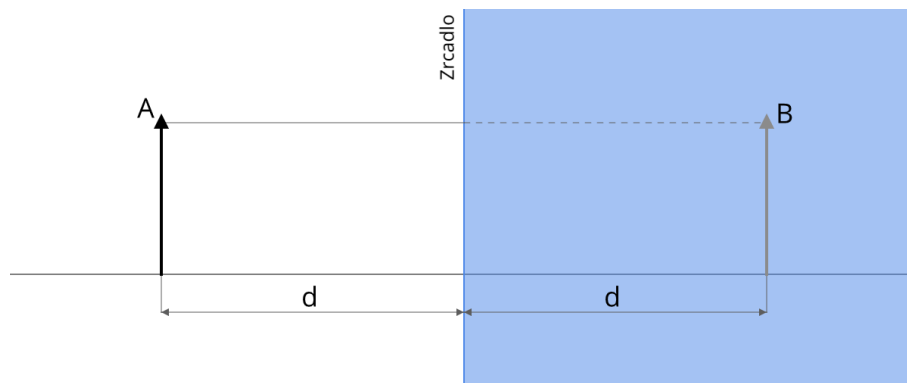
Duté zrcadla se v antice používala k zapalování ohně, jelikož soustředila dopadající sluneční paprsky zhruba do jednoho bodu, *ohniska (fokus)*. Termín fokus zavedl německý matematik, astronom a evangelický teolog JOHANNES KEPLER v roce 1604. [6]

Dopadají-li na duté zrcadlo *paraxiální paprsky*⁴ rovnoběžné s optickou osou, pak odražené paprsky procházejí ohniskem. Nejedná-li se o paraxiální rovnoběžné paprsky, budou procházet pouze okolím ohniska. Za takové rovnoběžné paprsky můžeme považovat přímé sluneční záření (vzhledem ke vzdálenosti Slunce).

Chod paprsku odraženého od kulového zrcadla můžeme určit ze zákona odrazu, jak můžeme vidět na obrázku 5. Pro geometrickou konstrukci se paprsek odráží od roviny tečné ke kulové ploše v bodě dopadu paprsku. Jedná-li se o praxiální paprsek, bude procházet ohniskem. Platí rovnice

$$|SV| = 2|VF|,$$

⁴Paprsky, které se šíří v blízkosti optické osy soustavy a svírají s ní malý úhel. V případě rovnoběžných paprsků se jedná o úhel 0°.

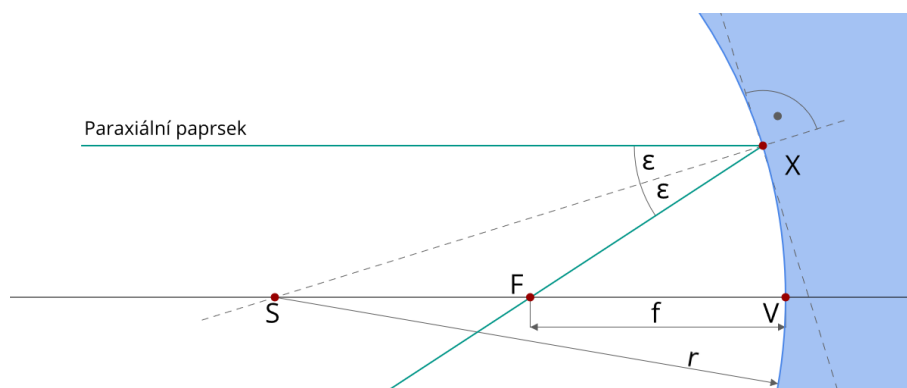


Obrázek 4 | Předmět A se nachází ve vzdálenosti d od zrcadla. Obraz B předmětu A vznikne ve vzdálenosti $2d$ od předmětu A , navíc je spojnice předmětu A a obrazu B kolmá na rovinu zrcadla. Geometricky se jedná o osovou souměrnost.

kterou můžeme psát ve tvaru

$$r = 2f, \quad (4)$$

kde r je poloměr sférické plochy a f je ohnisková vzdálenost zrcadla.



Obrázek 5 | Odraz paraxiálního světelného paprsku od dutého zrcadla. r je poloměr zrcadla, f je ohnisková vzdálenost, o je optická osa, S , F , V jsou po řadě střed, ohnisko a vrchol zrcadla. Obrázek je překreslen a upraven z [6].

1.3.1. Zobrazovací rovnice sférického zrcadla

Odvození zobrazovací rovnice sférického zrcadla bude užitečné pro odvození podobné rovnice pro lámavé plochy. Úvaha v tomto a následujícím odstavci je přebrána z [6]. Mějme paraxiální paprsek, který svírá s optickou osou úhel α a odražený paprsek svírající s toutéž osou úhel β , viz obrázek 6. Úhel dopadu ε je měřený od kolmice k tečně kulové plochy v bodě dopadu X .

Z obrázku 6 a trojúhelníků AXS a AXB jsou zřejmé rovnosti

$$\begin{aligned} \alpha + \varepsilon &= \psi, \\ \alpha + 2\varepsilon &= \beta, \end{aligned}$$

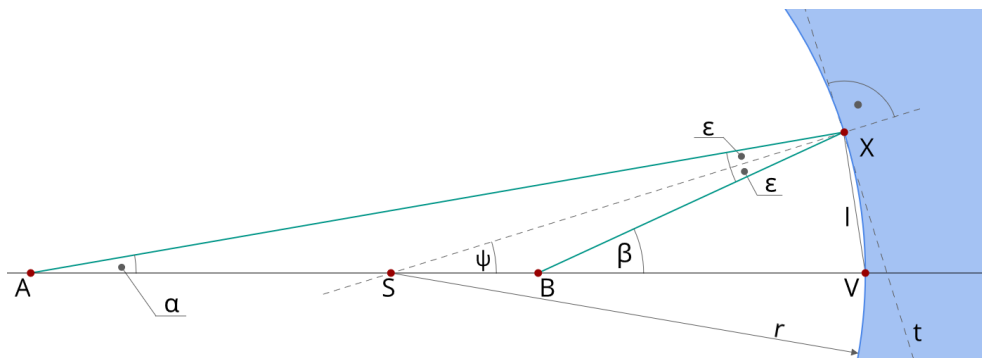
po úpravách dostaneme

$$\alpha + \beta = 2\psi. \quad (5)$$

Jelikož pracujeme s malými úhly, můžeme zavést aproximaci

$$\sin \xi \approx \xi$$

a předpokládat, že trojúhelníky AXV , SXV a BXV jsou pravoúhlé.



Obrázek 6 | Chod paprsku a odraženého paprsku soustavou dutého zrcadla k odvození zobrazovací rovnice sférického zrcadla. Obrázek je překreslen a upraven z [6].

Z toho plyne

$$\alpha \approx \frac{|XV|}{|AV|}, \psi \approx \frac{|XV|}{|SV|}, \beta \approx \frac{|XV|}{|BV|}.$$

Po dosazení do 5 máme

$$\frac{|XV|}{|AV|} + \frac{|XV|}{|BV|} = \frac{2|XV|}{|SV|},$$

a po úpravách

$$\frac{1}{|AV|} + \frac{1}{|BV|} = \frac{2}{|SV|}. \quad (6)$$

Vztah výše označujeme jako *zobrazovací rovnici sférického zrcadla*. Vztah platí pouze pro paraxiální paprsky.

Zobrazovací rovnici 6 můžeme dle rovnosti 4 přepsat na tvar

$$\frac{1}{|AV|} + \frac{1}{|BV|} = \frac{1}{f}. \quad (7)$$

Tuto rovnici 7 nazýváme *Gaussova zobrazovací rovnice*; publikována byla v roce 1825 HENRYM CODDINGTONEM ve spise *An Elementary Treatise on Optics*. [6]

1.3.2. Zvětšení obrazu sférickým zrcadlem

Pomocí zobrazovací rovnice 6 dokážeme najít obraz osového bodu. Nyní se budeme zabývat konstrukcí bodu mimoosového.

Mějme osový bod A a jeho obraz B v dutém zrcadle. Dále vytvořme bod C rotací bodu A kolem bodu S o malý úhel φ . Spojnice bodu B a středu sférické plochy S bude tvořit novou pomocnou optickou osu. Bod C je tedy bod pomocné optické osy a dle zobrazovací rovnice 6 najdeme obraz D bodu C.

Jelikož je úhel φ malý, můžeme považovat AC a BD za kolmice k původní optické ose SV a psát

$$\tan \varphi = \frac{|AC|}{|AS|} = \frac{|BD|}{|SB|}.$$

po úpravě dostaneme

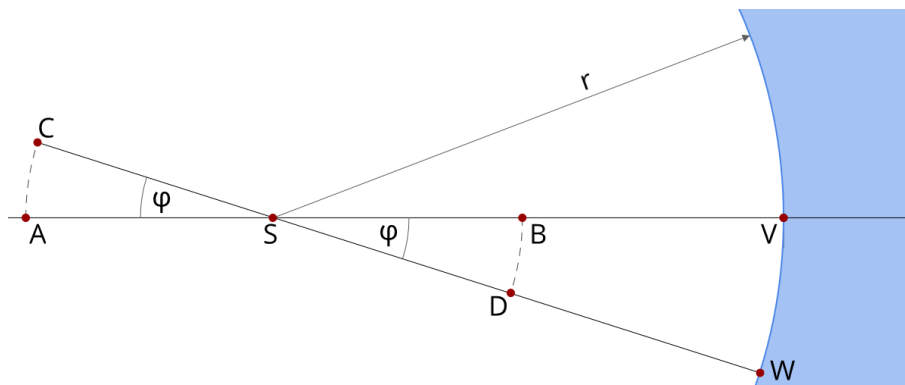
$$\frac{|BD|}{|AC|} = \frac{|SB|}{|AS|}.$$

Velikosti úseček $|AS|$ a $|SB|$ nahradíme ze vztahů

$$\begin{aligned} |SB| &= |SV| - |BV|, \\ |AS| &= |AV| - |SV| \end{aligned}$$

a dostaneme

$$\beta = \frac{|BD|}{|AC|} = \frac{|SV| - |BV|}{|AV| - |SV|}, \quad (8)$$



Obrázek 7 | Obraz osového a mimoosového bodu pro určení příčného zvětšení zobrazovacího zrcadla. Obrázek je překreslen a upraven z [6].

pomocí zobrazovací rovnice 6 upravíme na konečný tvar

$$\beta = \frac{|BD|}{|AC|} = -\frac{|BV|}{|AV|},$$

kde β je příčné zvětšení sférického zrcadla.

1.4. Zobrazení lámavými plochami

Abychom mohli studovat chod světelného paprsku optickou soustavou složenou z lámavých ploch⁵, musíme nejprve zavést konvenci pro znaménka veličin a vysvětlit chování světelného paprsku procházejícího jednou lámavou plochou.

V simulaci se budeme zabývat pouze kulovými lámavými plochami, případně rovinnou plochou, proto se omezíme na tyto případy i v následujícím výkladu.

1.4.1. Znaménková konvence pro lámavé plochy

Aby nedošlo k nedorozumění při určování znaménka veličin zavádí se takzvaná znaménková konvence:

1. Předpokládá se, že světlo přichází od levé strany k pravé; tento směr se považuje za kladný.
2. Je-li bod A , resp. A' vpravo (vlevo) od vrcholu V , je sečná vzdálenost a , resp. a' kladná (záporná).
3. Poloměr křivosti r lámavé plochy je kladný (záporný), je-li střed křivosti S vpravo (vlevo) od vrcholu V .
4. Úhel σ , resp. σ' , má znaménko podílu $\frac{h}{a}$, resp. $\frac{h}{a'}$, přičemž výška dopadu h je kladná (záporná), je-li bod dopadu X nad (pod) optickou osou.
5. Úhel dopadu ε měříme, jak jsme již uvedli, od paprsku ke kolmici; ve směru hodinových ručiček je úhel kladný, v opačném případě záporný.
6. Úhel lomu ε' má znaménko totožné s úhlem dopadu.”[7]⁶

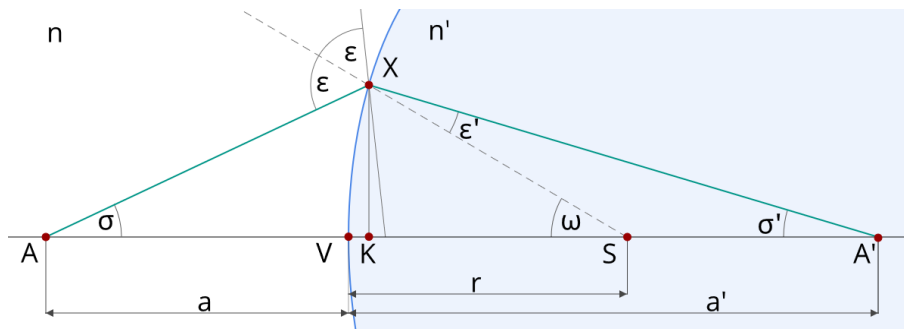
Tato konvence se liší od konvence běžně využívané na středních školách nebo v knize [6], další literatura však používá uvedenou konvenci. Stejná konvence se využívá při výuce optiky na Přírodovědecké fakultě Univerzity Palackého v Olomouci.

1.4.2. Chod paprsku jednou kulovou lámavou plochou

Mějme paprsek svírající úhel σ s optickou osou, který se ve vzdálenosti h od optické osy láme na sférické lámavé ploše a protíná optickou osu v bodě A' , viz obrázek 8.

⁵Dvojice těchto lámavých ploch budou v simulaci představovat tlusté čočky.

⁶V citaci byly zaměněny symboly bodů, po řadě X, X', S, C, J byly nahrazeny symboly A, A', V, S, X , aby text citace odpovídal značení ve zbytku práce.



Obrázek 8 | Schéma chodu paprsku soustavou jedné sférické lámavé plochy. Obrázek je překreslen a upraven z [7].

Následující úlohou je určit *obrazovou vzdálenost* a' a *dopadový úhel* σ' ze znalosti *předmětové vzdálenosti* a a úhlu σ .

Z trojúhelníku AXS a sinové věty je vidět vztah

$$\frac{r}{\sin \sigma} = \frac{|a| + r}{\sin(180 - \varepsilon)},$$

po úpravě dostaneme

$$\sin \varepsilon = \frac{a - r}{r} \sin \sigma. \quad (9)$$

Vztah pro určení $\sin \varepsilon'$ získáme ze zákona lomu 3 jako

$$\sin \varepsilon' = \frac{n}{n'} \sin \varepsilon. \quad (10)$$

Velikost úhlu dopadu σ' vyplývá z vlastnosti vrcholových úhlů, dostaneme:⁷

$$\sigma' = \sigma + \varepsilon - \varepsilon'. \quad (11)$$

Konečně vyjádříme obrazovou vzdálenost

$$a' = r + r \frac{\sin \varepsilon'}{\sin \sigma'}, \quad (12)$$

kde jsme využili sinovou větu v trojúhelníku SXA' .

Nakonec vyjádříme výšku dopadu h . Nejprve zjistíme, že $\omega = \varepsilon + \sigma$ ⁸, pak už z trojúhelníku KXS určíme

$$h = r \sin(\varepsilon + \sigma). \quad (13)$$

V předchozím jsme neošetřili případ paprsku rovnoběžného s optickou osou. V tomto případě je dle [7]⁹ nutné nahradit rovnici 9 rovnicí

$$\sin \varepsilon = \frac{h}{r}. \quad (14)$$

Úlohu chodu paprsku přes jednu kulovou lámavou plochu máme splněnu, v dalším budeme stopovat paprsek přes konečné množství kulových lámavých ploch. Odvození v tomto odstavci bylo provedeno dle [7].

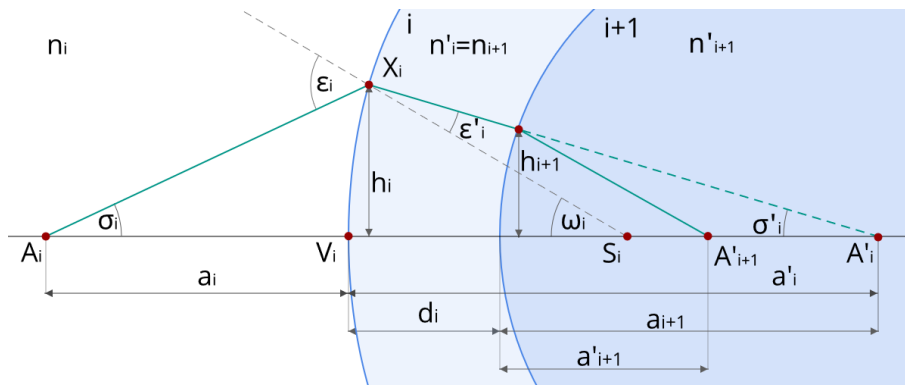
1.4.3. Stopování paprsku skrze konečné množství sférických lámavých ploch

Při větším počtu lámavých ploch budeme postupovat po řadě. Nejprve určíme, jak se zlomí paprsek po průchodu první lámavou plochou, pak po průchodu druhou, třetí atp. Představme si situaci z obrázku 9.

⁷Je nutné zohlednit zavedenou znaménkovou konvenci z 1.4.1. Pokud bychom tak neučinili, dostali bychom $\sigma' = \varepsilon - \sigma - \varepsilon'$

⁸Opět je nutné zohlednit znaménkovou konvenci.

⁹A také z představy rovnoběžného paprsku na obrázku 8.



Obrázek 9 | Dvě po sobě jdoucí lámavé plochy. Všimněte si rozdílu mezi a'_i a a_{i+1} . Obrázek je překreslen a upraven z [7].

Platí, že úhel dopadu paprsku po průchodu i -tou plochou je roven úhlu, pod kterým dopadá na $i+1$ -tou lámavou plochu. Při zachování značení platí

$$\sigma_{i+1} = \sigma'_i. \quad (15)$$

Obdobný vztah platí pro indexy lomu n a n' . Můžeme psát

$$n_{i+1} = n'_i. \quad (16)$$

Ovšem pro předmětovou a obrazovou vzdálenost již neplatí, neboť tyto vzdálenosti měříme vždy od vrcholu lámavé plochy. Je-li vzdálenost mezi vrcholy i -té a $i+1$ -té lámavé plochy d_i , můžeme psát

$$a_{i+1} = a'_i - d_i. \quad (17)$$

Tím máme problém trasování paprsku konečným počtem lámavých ploch vyřešen.

1.4.4. Paprsky v paraxiálním prostoru

Zavedme si nejprve pojem *ideálního zobrazení*. Havelka jej definuje v [7] jako zobrazení, při kterém se všechny paprsky vycházející z jednoho bodu, po průchodu optickou soustavou, opět protnou v jednom bodě. Havelka tamtéž pokračuje konstatováním, že ideální obraz v případě odrazu získáme pomocí zrcadla ve tvaru rotačního elipsoidu nebo rotačního paraboloidu, dle polohy zobrazovaného předmětu. Je-li předmět v konečné vzdálenosti, je nutné zrcadlo ve tvaru rotačního elipsoidu, je-li ve vzdálenosti nekonečné, pak je nutné použití zrcadla ve tvaru rotačního paraboloidu. Konstruujeme-li obraz pomocí lomu, vzniká ideální obraz ve vzdálenosti $\frac{n'}{n}r$ u kulové plochy. Veličiny n' , n a r označují po řadě index lomu prostředí za lámavou plochou,¹⁰ před lámavou plochou a poloměr lámavé plochy.

“Čočka zprostředkuje téměř ideální zobrazení jen velmi malého plošného elementu v blízkosti optické osy, a to paprskovými svazky velmi úzkými; prostor, v němž tyto zobrazující svazky probíhají, sluje *prostor paraxiální*.”[7]

Tedy paraxiální prostor je takový prostor, ve kterém se šíří paraxiální paprsky.¹¹ Pro tyto malé úhly platí přibližné vztahy

$$\begin{aligned} \sin \xi &\approx \xi, \\ \tan \xi &\approx \xi, \\ \cos \xi &\approx 1. \end{aligned} \quad (18)$$

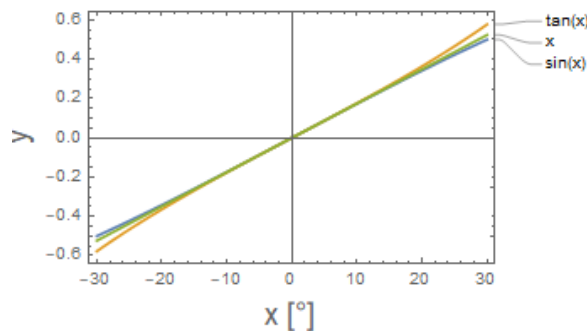
Jak můžeme vidět v grafu 10, jsou rozdíly v blízkosti nuly minimální.¹²

Nyní můžeme rovnice 9 až 13 pomocí vztahů 18 upravit do tvaru

¹⁰Respektujeme-li zavedenou znaménkovou konvenci 1.4.1 a tedy i směr šíření paprsku zleva doprava.

¹¹Definice pojmu v 1.3

¹²Během výuky jsme si uváděli interval, ve kterém můžeme vztahy použít, $\langle -5^\circ; 5^\circ \rangle$.



Obrázek 10 | Porovnání hodnot funkcí $\sin \xi$, $\tan \xi$ a ξ v blízkosti nuly. Graf byl vytvořen ve Wolfram Mathematica.

$$\begin{aligned}
 \varepsilon &= \frac{a-r}{r}\sigma, \\
 \varepsilon' &= \frac{n}{n'}\varepsilon, \\
 \sigma' &= \sigma + \varepsilon - \varepsilon', \\
 a' &= r + r\frac{\varepsilon'}{\sigma'}, \\
 h &= r(\sigma + \varepsilon).
 \end{aligned} \tag{19}$$

Rovnice 19 se shodují s rovnicemi uvedenými v [7].

Vztah 14, který ošetřuje dopad paprsku rovnoběžného s optickou osou nabývá tvaru

$$\varepsilon = \frac{h}{r}. \tag{20}$$

1.4.5. Zobrazovací rovnice lámavé plochy, příčné zvětšení a ohniska

Při odvozování zobrazovací rovnice pro lámavou plochu, budeme postupovat obdobně jako při odvozování zobrazovací rovnice pro sférické zrcadlo. Úvaha v této části je přebrána z [6]. Budeme vycházet z obrázku 8, odvození provedeme v paraxiálním prostoru.

Vezmeme zákon lomu 3 a použijme aproximace 18. Dostaneme ¹³

$$n\varepsilon = n'\varepsilon'. \tag{21}$$

Z trojúhelníků AXS a SXA' získáme vztahy

$$\begin{aligned}
 \varepsilon &= \omega + \sigma, \\
 \omega &= \varepsilon' + \sigma'.
 \end{aligned} \tag{22}$$

Úpravou vztahů 21 a 22 dostaneme rovnost

$$n\sigma + n'\sigma' = (n' - n)\omega. \tag{23}$$

Vzhledem k tomu, že odvození probíhá v paraxiálním prostoru, můžeme považovat trojúhelníky AXV , VXS a VXA' za pravoúhlé a tedy platí přibližné vztahy ¹⁴

$$\sigma \approx \frac{|VX|}{a}, \omega \approx \frac{|VX|}{r}, \sigma' \approx \frac{|VX|}{a'}. \tag{24}$$

Jejich užitím a vztahem 23 dostaneme hledanou zobrazovací rovnici ¹⁵

$$\frac{n'}{a'} - \frac{n}{a} = \frac{n' - n}{r}. \tag{25}$$

¹³Označení odpovídá obrázku 8 v tomto případě i v celé podkapitole.

¹⁴S použitím aproximací 18.

¹⁵Kvůli znaménkové konvenci je na levé straně mínus. Můžeme si všimnout podobnosti se zobrazovací rovnicí sférického zrcadla 6.

Druhou možností odvození zobrazovací rovnice je vyloučení úhlů z prvních čtyř vztahů 19. [7]
Postup odvození příčného zvětšení lámavé plochy je téměř shodný s postupem odvození příčného zvětšení sférického zrcadla. Proto rovnou uvádíme vztah,

$$\beta = \frac{y'}{y} = \frac{a' - r}{a + r}, \quad (26)$$

kde jsou y a y' vzdálenosti zobrazovaného bodu a zobrazeného bodu od optické osy, a který je obdobou vztahu 8.

Následnou úpravou rovnice 26 pomocí rovnice 25 dostaneme další vztah pro příčné zvětšení ¹⁶

$$\beta = \frac{na'}{n'a}. \quad (27)$$

Dále se definuje pojem *lámavost*, nebo-li *optická mohutnost* a zavádí se vztahem ¹⁷

$$\phi = \frac{n' - n}{r}. \quad (28)$$

Jednotkou je zřejmě m^{-1} , často se jednotka označuje jako dioptrie, značka jednotky je D nebo dpt. [6]

Zavedme si termín *předmětové*, resp. *obrazové*, *ohnisko* lámavé plochy. ¹⁸ Jedná se o osové body F , resp. F' , pro které je příčné zvětšení $\beta = \infty$, resp. $\beta = 0$. [6] Je nutno upozornit, že obrazem bodu F není bod F' a obráceně. Stejně jako u sférických zrcadel se i u sférických lámavých ploch zavádí termín *ohnisková vzdálenost*. Přesněji *předmětová* resp. *obrazová* ohnisková vzdálenost. Jedná se o vzdálenost měřenou od vrcholu lámavé plochy po předmětové resp. obrazové ohnisko. Používá se značení f , resp. f' . I pro tyto pojmy zůstává v platnosti znaménková konvence 1.4.1.

Pomocí pojmu ohniskových vzdáleností můžeme přepsat rovnici 25 na tvar

$$\frac{f}{a} + \frac{f'}{a'} = 1,$$

kde

$$f = \frac{nr}{n' - n}, \quad (29)$$

$$f' = \frac{n'r}{n' - n}. \quad (30)$$

1.4.6. Kardinální body optických soustav a jejich určení, úhlové zvětšení

Kardinálními body můžeme nazvat *předmětový*, resp. *obrazový*, *hlavní bod*, značíme H , resp. H' . V tomto případě se již jedná o sdružené body. Tedy H se skrz optickou soustavu zobrazí do H' . Oba hlavní body jsou osové body a mají příčné zvětšení rovné $\beta = 1$. [6]

V podkapitole 1.4.5, tj. minulý odstavec, jsme zavedli pojmy ohniska a ohniskové vzdálenosti. Pro optickou soustavu složenou z konečného počtu lámavých ploch zůstává definice ohniska nezměněná, ovšem nepatrně se mění pojem ohniskové vzdálenosti. U soustav lámavých ploch měříme předmětovou ohniskovou vzdálenost od předmětového hlavního bodu a obrazovou od obrazového hlavního bodu. [7] V případě, že máme pouze jednu lámavou plochu, oba hlavní body splynou s vrcholem lámavé plochy. Body F a F' také nazýváme kardinálními body.

Problém určení obrazové i předmětové ohniskové vzdálenosti je triviální. Chceme-li určit obrazovou ohniskovou vzdálenost soustavy lámavých ploch, budeme soustavou stopovat paprsek rovnoběžný s optickou osou, tedy zvolíme $a_1 = -\infty$. [7] Jako poslední obrazovou vzdálenost dostaneme právě vzdálenost obrazového ohniska a poslední lámavé plochy. Poté již snadno do počteme polohu ohniska. Chceme-li určit předmětové ohnisko, otočíme celou soustavu. ¹⁹

Obrazovou ohniskovou vzdálenost určíme ze vztahu

$$f' = \frac{a'_1 a'_2 \dots a'_j}{a_2 a_3 \dots a_j}, \quad (31)$$

¹⁶Oproti [6] chybí před zlomkem mínus. Provedli jsme tutu změnu, aby vztah korespondoval se znaménkovou konvencí 1.4.1.

¹⁷Jedná se o pravou stranu zobrazovací rovnice lámavé plochy 25

¹⁸Definice platí stejně i pro složitější optické soustavy.

¹⁹První lámavá plocha bude poslední, druhá bude předposlední, atp.

kde jednotlivé předmětové a obrazové vzdálenosti odpovídají zobrazování předmětu z nekonečna, tedy $a_1 = -\infty$. [7]

Jelikož je ohnisková vzdálenost vzdáleností mezi hlavním bodem a ohniskem, pak ze znalosti ohniska určíme polohu hlavního bodu. Zavedme ještě pojem *předmětové*, resp. *obrazové*, *hlavní vzdálenosti* jako vzdálenost mezi první lámavou plochou a předmětovým hlavním bodem, resp. poslední lámavou plochou a obrazovým hlavním bodem.

Abychom si mohli nadefinovat poslední dvojici kardinálních bodů, musíme zavést pojem *úhlové zvětšení* γ . Dle [7] je definováno vztahem

$$\gamma = \frac{n_1}{n'_j} \frac{1}{\beta}, \quad (32)$$

kde n_1, n'_j jsou indexy lomu před první lámavou plochou a za poslední.²⁰

Šestici kardinálních bodů uzavírají *předmětový* a *obrazový uzlový bod*. Jedná se opět o sdružené body. Dva osové body se nazývají *předmětový* a *obrazový uzlový bod*, mají-li úhlové zvětšení $\gamma = 1$.

Význam kardinálních bodů spočívá v charakterizaci optické soustavy. Optickou soustavu plně charakterizují ohniskové body a jedna z dvojic uzlových nebo hlavních bodů. [7]

1.4.7. Určení zvětšení optické soustavy a polohy hlavních bodů

Velikost příčného zvětšení celé soustavy určíme jako součin zvětšení zobrazení jednotlivými lámavými plochami, tedy

$$\beta = \beta_1 \beta_2 \dots \beta_j. \quad (33)$$

Dosažením vztahu 27 do 33 dostaneme²¹

$$\beta = \frac{n_1}{n'_j} \frac{a'_1 a'_2 \dots a'_j}{a_1 a_2 \dots a_j} = \frac{n_1}{n'_j} \frac{h_1}{h_j} \frac{a'_j}{a_1}. \quad (34)$$

1.4.8. Osové zvětšení optické soustavy

Mějme dva body A a B v předmětovém prostoru. Označme vzdálenost $|AB| = b_1 - a_1 = g_1$ a vzdálenost $|A'B'| = b'_1 - a'_1 = g'_j$. Nyní můžeme osové zvětšení dle [7] definovat vztahem

$$\alpha = \frac{g'_j}{g_1} = \frac{n'_j}{n_1} \beta_a \beta_b, \quad (35)$$

kde β_a a β_b jsou příčná zvětšení bodů A a B .

1.4.9. Čočky

V simulaci budeme čočkou rozumět dvojici za sebou jdoucích lámavých ploch, přičemž index lomu za první lámavou plochou bude odpovídat indexu lomu před druhou lámavou plochou. Další informace o čočkách nejsou pro tvorbu modelu v programu Wolfram Mathematica nezbytné. Více informací naleznete v [6, 7].

Pro spoustu aplikací je vhodné využít maticový formalismus.²² Více o tomto způsobu trasování naleznete v [8].

²⁰Nadále budeme písmenem "j" označovat poslední lámavou plochu. Veličiny s indexem "j" se budou k poslední lámavé ploše vztahovat, například n_j a n'_j budou značit index lomu před a za poslední lámavou plochou.

²¹ Z první a poslední rovnice z 19, dle [7], dostaneme

$$h = a\sigma = a'\sigma',$$

vztah je platný pro i -tou i $i+1$ -tou lámavou plochu. Vytvoříme poměr a z rovnosti $\sigma'_i = \sigma_{i+1}$ dostaneme

$$\frac{h_i}{h_{i+1}} = \frac{a'_i \sigma'_i}{a_{i+1} \sigma_{i+1}} = \frac{a'_i}{a_{i+1}},$$

z toho již

$$\frac{h_i}{h_j} = \frac{a'_1 a'_2 \dots a'_{j-1}}{a_2 a_3 \dots a_j}.$$

²²Během tvorby simulace jsme tento formalismus nadšeně využívali, protože se zdál jako velké zjednodušení, bohužel se ukázalo, že je tento přístup pro náš projekt nevhodný. Řečeno českým rčením, jednalo se o použití kanónu na vrabce.

2. Představení prostředí Wolfram Mathematica

První verze prostředí Wolfram Mathematica bylo vydáno Stephenem Wolframem, britským fyzikem a autorem knihy *A New Kind Of Science*, 23. června 1988. Na jaře roku 2019 byla vydána již 12. verze prostředí. [9]

Ve Wolfram Documentation Center je jazyk Wolfram²³ představován:

“The Wolfram Language is a highly developed knowledge-based language that unifies a broad range of programming paradigms and uses its unique concept of symbolic programming to add a new level of flexibility to the very concept of programming”. [10]

V této kapitole si vysvětlíme základní konstrukce a funkce prostředí Wolfram Mathematica použité v modelu popsaném v kapitole 3. Možnosti prostředí jsou samozřejmě mnohem širší.

Prostředí Wolfram Mathematica má velmi podrobně propracovanou dokumentaci [10] a také existuje řada publikací, které jasně vysvětlují jak s jazykem i prostředím pracovat. Doporučujeme zejména publikaci Stephena Wolframa *An elementary introduction to the Wolfram language*. [1]

Prostředí Wolfram Matematika je dostupné jak pro počítače s Windows, MacOS i Linux, tak skrze webový prohlížeč. Jazyk Wolfram je skriptovací jazyk, tedy kód můžeme spouštět po částech a zdefinované výrazy jsou platné do vypnutí prostředí nebo do jejich smazání a to napříč *notebooky*. V notebooku můžeme najít tři druhy řádků:

- řádky pro zadání obyčejného textu, pro zřehlednění kódu;
- vstupy pro prostředí Mathematica, které jsou zpracovávány, evaluovány;
- výstupy prostředí po zpracování našeho kódu.

2.1. Symbolické výrazy

“At the core of the Wolfram Language is the foundational idea that everything—data, programs, formulas, graphics, documents—can be represented as symbolic expressions. And it is this unifying concept that underlies the Wolfram Language’s symbolic programming paradigm, and makes possible much of the unique power of the Wolfram Language and the Wolfram System.”[10]

Všechny výrazy mají stejnou strukturu `Name[Parametr1, Parametr2, ...]`, Jméno označujeme jako *head*, soubor všech parametrů jako *body*. V některých případech se používá alternativní značení, třeba u sčítání, násobení atp.²⁴ Výraz $4 + 6$ se může zapsat jako `Plus[4,6]`. Výrazům můžeme přiřadit obsah několika způsoby, uvedeme dva:

1. `random1 = RandomInteger[{1,6}];`
2. `random2 := RandomInteger[{1,6}].`

Kód v prvním řádku uloží pod výraz `random1` náhodné číslo od jedné do šesti a kdykoli jej zavoláme, vždy nám vrátí stejné číslo.²⁵ Zatímco výraz `random2` nám při každém zavolání vrátí jiný výsledek, tedy výraz se pokaždé znovu zpracuje.

Označení *funkce* budeme používat vždy pro výrazy zdefinované pomocí `:=`. Například zavedeme funkci `random3`, která vrátí náhodné celé číslo ze zadaného intervalu.

```
random3[x_, y_] := RandomInteger[{x,y}]
```

Zavádíme-li funkci, jména jejich parametrů musí následovat podtržítka.

²³Jazyk používaný v prostředí Mathematica i dalších produktech společnosti Wolfram LLC.

²⁴Také u `List[]`, `Association[]`, `Rule[]`, `Numeric[]`, `ReplaceAll[]`.

²⁵Pokud výraz `random1` nepřepíšeme jinou hodnotou.

2.2. List

V jazyce Wolfram jsou zavedeny také seznamy výrazů. Chceme-li vytvořit seznam tří náhodných celých čísel z intervalu $< 1; 10 >$ a uložit jej do proměnné `threeRandom` zpracujeme kód níže.

```
threeRandom = List[random3[1,10], random3[1,10], random3[1,10]]
```

Stejný výraz lze zapsat pomocí `{}`.

```
threeRandom = {random3[1,10], random3[1,10], random3[1,10]}
```

Nyní můžeme vybrat třeba druhý prvek ze seznamu, kterému budeme dále říkat list, zavoláním výrazu `threeRandom[[2]]`. Existuje celá řada built-in funkcí, které s listy pracují. Uvedme:²⁶

- `First[a]` → vrátí první hodnotu listu `a`;
- `Last[a]` → vrátí poslední hodnotu listu `a`;
- `Length[a]` → vrátí délku listu `a`;
- `Prepend[a, A]` → vrátí list `a` s připojeným výrazem `A` na začátku;
- `Append[a, A]` → vrátí list `a` s připojeným výrazem `A` na konci;
- `Join[a, b, ...]` → vrátí list vzniklý spojením všech listů uvedených v argumentu;
- `Sort[a, p]` → vrátí přeuspořádaný list, parametr `p` je řadící funkce.²⁷

Jako všechno v prostředí Mathematica patří i list mezi výrazy, ze kterých můžeme složit další list. Interpretací listu složeného z listů je třeba tabulka nebo matice. Představme si nyní takovýto složený list jako matici. Známe operaci transponování matic a tu nám v Mathematice obstarává funkce `Transpose[]`.

Další funkcí pracující s listy zanořenými do listů je `Flatten[]`. Tato funkce listy doslova zplošťuje. Uvedme příklad: z listu `{{A, B}, {C, A}, C}`, vytvoří list `{A, B, C, A, C}`. Pomocí dalších parametrů můžeme nastavit "zploštění" detailněji.

2.3. Association List

Pro reprezentaci reálných objektů, které mají specifické vlastnosti je vhodné využít strukturu, která nám po zavolání jména struktury s argumentem charakterizujícím danou vlastnost, vrátí míru dané vlastnosti. Například každá lámavá plocha má určený poloměr a polohu v optické soustavě. Je vhodné mít jednu strukturu, která reprezentuje lámavou plochu a na ní vlastnosti poloměr a poloha. Takovou strukturou je právě association list. Zdefinujme si association list z příkladu.

```
refractingSurface = Association[Rule["radius", -5], Rule["position", 10]]
```

Nyní již můžeme zavolat `refractingSurface["radius"]` a dostaneme výsledek `-5`.

Použili jsme funkce `Association[]` a `Rule[]`. Podívejme se blíže na funkci `Rule[]`, tato funkce vytváří takzvaná pravidla, která výrazu přiřadí hodnotu ještě nezadefinovaným způsobem. Častěji se vyjadřuje pomocí `" → "` jako `"position" → 10`.

Druhou použitou funkcí je `Association[]`, díky které nám jsou vráceny správné hodnoty po zavolání příslušného parametru, těmto parametrům se říká klíče. Funkci `Association[...]` můžeme zapsat ve tvaru `< |...| >`.

Souhrnně. Předchozí kód lze zapsat přehledněji.

```
refractingSurface = <|"radius" → -5, "position" → 10|>
```

²⁶Pod názvy `a`, `b` atp. si představujeme libovolný list a pod `A`, `B` atp. libovolný výraz.

²⁷Příkladem řadící funkce je `Greater`, která seřadí prvky od největšího. V tomto případě je vhodné použití tzv. *pure function*, v jiných jazycích nazývané anonymní. Příkladem je funkce `#1>#2&`, která je ekvivalentní funkci `Greater`.

2.4. Logické výrazy

Logické operace stejně jako vše jsou v programu Mathematica implementovány jako výraz. Vrací vždy hodnotu `True` nebo `False`. Výraz `Or[]`, resp. `And[]`, můžeme psát jako `| |`, resp. `&&`. Testování rovnosti se provádí pomocí `Equal[]`, používá se `i ==`.

Implementovány jsou i funkce ověřující typ výrazu. Například funkce `NumberQ[A]` vrátí `True`, pokud je `A` číslo,²⁸ v opačném případě `False`.

2.5. Iterace a podmínky

V prostředí Mathematica je zabudována klasická podmínka `If[]` a cykly `For[]`, `Do[]`, `While[]`. Během tvorby modelu budeme nejvíce využívat cyklus `Table[]`.

Cyklus `Table[expression[i], {i, imin, imax}`, provede výraz `expression[i]` pro každou celou hodnotu `i` z intervalu `< imin; imax >` a vrátí souhrnný list. Například jednoduchý výraz `Table[i, {i, 1, 5}]`, vrátí list `{1, 2, 3, 4, 5}`.

2.6. Lokální proměnné a volitelné parametry funkcí

Jelikož jsou všechny výrazy v prostředí Mathematica globální, používá se funkce `Module[]`, pro vytváření výrazů lokálních. Tedy výrazů, které existují jen v těle funkce `Module[]`. Uvedeme si příklad.

```
Module[{a = 5}, a]
```

Kód výše nám vrátí: 5. Jestliže, ale výraz `a` zavoláme mimo tělo funkce `Module[]`, Mathematica nám vrátí `a`. Funkce `Module[]` se hojně využívá pro definování nových funkcí.

Při definování nových funkcí je také vhodné nastavit volitelné parametry. V jazyce Wolfram jsou k tomu připraveny funkce `Options[]`, `OptionPattern[]` a `OptionValue[]`. Ukážeme si jak fungují na příkladu.

```
Options[f] = {"Option" → "Optional parameter."}
f[OptionPattern[]] := OptionValue["Option"]
```

Po zavolání funkce `f[]` nám prostředí vrátí text `"Optional parameter."`, zatímco když zavoláme funkci `f["Option" → "I changed optional parameter!"]`, prostředí vrátí text `"I changed optional parameter!"`.

2.7. Matematické funkce

"The Wolfram System's ability to deal with symbolic expressions, as well as numbers, allows you to use it for many kinds of mathematics."^[10]

V jazyce Wolfram můžeme stejně jako s matematickými proměnnými pracovat i se symboly, počítat derivace, limity, řešit rovnice, exaktně nebo numericky. K těmto výpočtům můžeme užít funkcí `D[g[x], x]`, `Limit[g[x], x → x0]`, `Solve[g[x] == 0, x]` a `NSolve[g[x] == 0, x]`.

K výpočtům limit se hodí rozšířit reálná čísla o nevlastní hodnoty, ty jsou implementovány výrazem `Infinity`, resp. `-Infinity`.

2.8. Plot, ContourPlot, Graphics a Show

Software Mathematica poskytuje množství funkcí určených k vizualizaci dat. My se omezíme jen na ty, které využijeme v našem modelu.

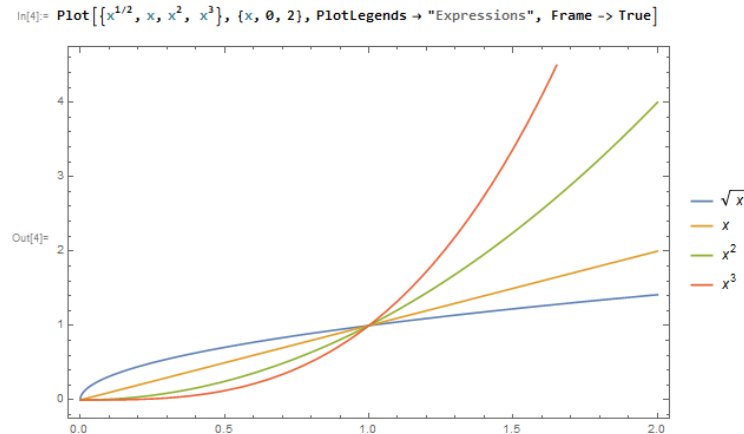
Funkce `Plot[g[x], {x, xmin, xmax}`, vykreslí graf funkce $y = g(x)$ v mezích od x_{min} do x_{max} . Tato funkce funguje pouze pro jednu nezávisle proměnnou a dvojrozměrný graf. Pro třírozměrný je vhodné použít funkce `Plot3D[]`. Při tvorbě modelu jsme potřebovali vykreslit i křivky s předpisem

²⁸Komplexní, reálné, racionální nebo celé.

typu $g(x) + h(y) = 0$, k tomu slouží následující funkce.

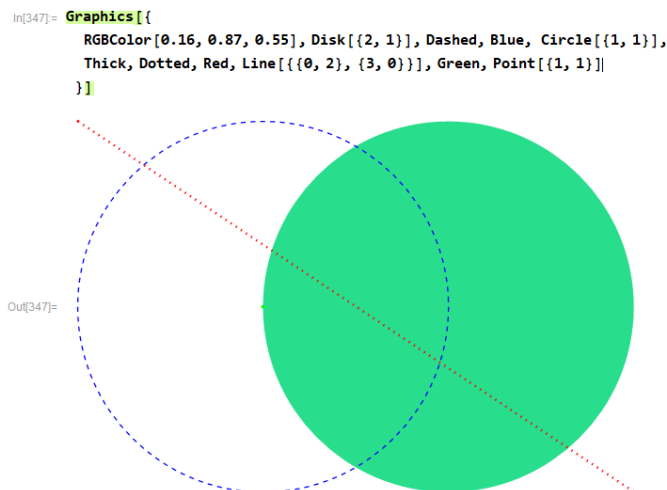
```
ContourPlot[g[x] + h[y] == 0, {x, x_min, x_max}, {y, y_min, y_max}]
```

Opět existuje i verze `ContourPlot3D[]`. Pro všechny čtyři zmíněné funkce existuje celá řada volitelných parametrů, které přidají nebo upraví osy, jejich popisy, styl a barvu vykreslení křivek, orámování grafu atp., viz obr. 11.



Obrázek 11 | Vykreslení grafů v prostředí Mathematica s užitím volitelných parametrů, které přidávají popisky průběhů funkcí a orámování.

Pro vykreslení některých objektů je mnohdy výhodnější použít funkci `Graphics[]`, která si jako argument bere list s výrazy, které upravují vlastnosti objektů a tyto objekty reprezentují. Příkladem je obrázek 12.



Obrázek 12 | Vykreslení pomocí funkce `Graphics[]`.

Výrazy reprezentující objekty jsou `Circle[]`, `Disk[]`, `Line[]` a `Point[]`.²⁹ Zbylé výrazy formují objekty, mění barvu, tvarují křivky atp.

Výrazy vrácené funkcemi `Plot[]`, `ContourPlot[]` a `Graphics[]` se souhrnně nazývají *graphics*. Funkce `Show[]` slouží ke skládání těchto *graphics* dohromady, což je velmi výhodné pro náš model. I tato funkce má celou řadu volitelných parametrů, uvedme `PlotRange` a `AxesOrigin`, které slouží k zobrazení požadovaného výřezu a umístění počátku os.

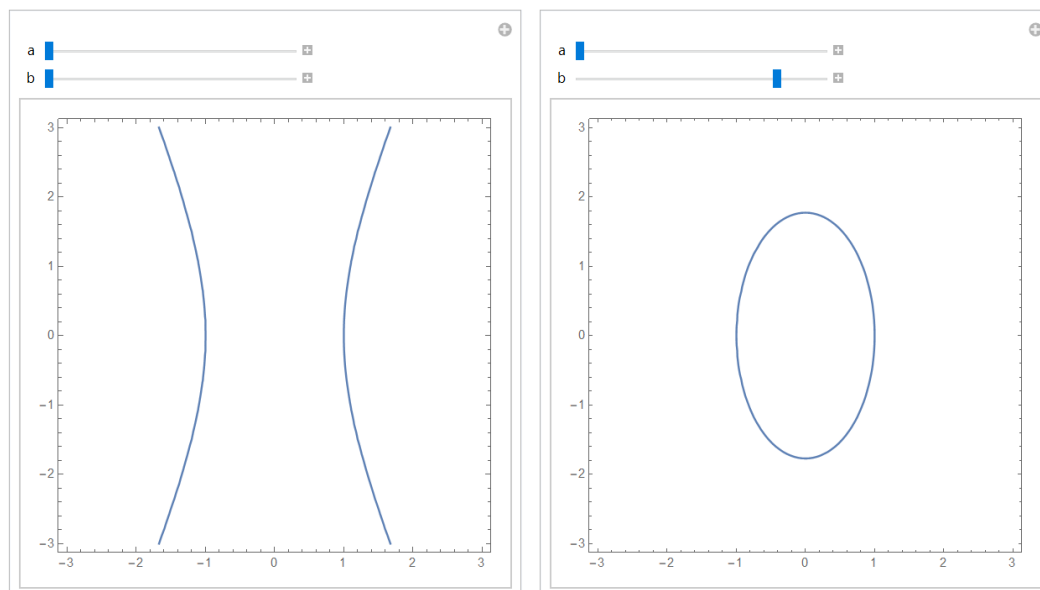
²⁹Během tvorby modelu jsme využili ještě `ArrowHeads[]` a `Arrow[]`, které znázorňují předměty a obrazy.

2.9. Manipulate, Control, FormObject, Failure a Dynamic

Funkce `Manipulate[]` přinese do našeho modelu dynamiku. Prvním parametrem je zpracovávaný kód, každý další pak nastavuje výrazy, které uživatel mění skrze vybrané prvky. Uvedeme příklad.

```
Manipulate[ContourPlot[ $\frac{x^2}{a} + \frac{y^2}{b} == 1$ , {x, -3, 3}, {y, -3, 3}], {a, 1, 5},  
{b, -5, 5}]
```

`Manipulate[]` výše vykreslí graf se dvěma slidery, jedním pro `a` a druhým pro `b`. V závislosti na jejich poloze se bude hyperbola měnit na elipsu, viz obr. 13.



Obrázek 13 *Funkce `Manipulate[]` měnící velikosti poloos elipsy nebo hyperboly, velikost hlavní poloosy je vždy kladná. Nalevo: velikost vedlejší poloosy je záporná. Napravo: velikost vedlejší poloosy se kladná.*

Slidery můžeme změnit na checkboxy, prázdná políčka atp. Jako druhý parametr funkce `Manipulate[]` můžeme použít i funkci `Control[]`,³⁰ kterou můžeme nastavovat dynamické prvky funkce `Manipulate[]` obdobně, ovšem funkci `Control[]` můžeme obalit do dalších funkcí a tím kupříkladu lépe formátovat výstup funkce `Manipulate[]`.

K vytvoření formuláře s větším počtem políček je vhodné použít funkci `FormObject[]`. Vytvoříme si ukázkový formulář se dvěma políčky do kterého budeme mít zadat jméno a telefonní číslo.

```
form = FormObject[{  
  {"Name", "Jméno:"} → <|"Interpreter" → String|>,  
  {"PhNum", "Telefonní číslo:"} →  
    <|"Interpreter" → Restricted["Integer", {0, 999999999}]|>  
}]
```

Funkce `FormObject` si bere za argument list složený z pravidel³¹, kde první parametr `Rule[]` je list složený ze jména políčka a textu zobrazovaného před políčkem, druhý parametr je association

³⁰Pokud `Control[]` zavoláme samostatně s argumentem `{x, -3, 3}`, vrátí nám pouze slider.

³¹Složený z `Rule[express1, express2]`, ten lze zapsat také ve tvaru `express1 → express2`.

list, ve kterém vidíme další pravidlo. Tentokrát "Interpreter" → `Restricted[...]`, který slouží k ověření obsahu políček zadaného uživatelem, viz obrázek 14. Pokud do políčka "PhNum" nezadáme celé číslo od 0 do 999999999, funkce `FormObject[]` skončí selháním, které se uživateli zobrazí jako "Enter a valid number". Uvnitř vrátí "Interpreter" funkci `FormObject[]` výraz `Failure[...]`, který funkce `FormObject[]` zobrazí uživateli.



Jméno:	<input type="text" value="Petr"/>	Jméno:	<input type="text" value="Petr"/>
Telefonní číslo:	<input type="text" value="Telefon?"/>	Telefonní číslo:	<input type="text" value="730854385"/>

Enter a valid number.

Obrázek 14 | Formulář pro zadání jména a telefonního čísla vytvořený pomocí `FormObject[]`. Na-
levo: špatně vyplněné telefonní číslo. Napravo: formulář je vyplněn správně.

Výrazy, zavedené jako první parametr funkce `Manipulate[]`, jsou automaticky dynamické. Tato dynamika se projevuje tím, že změníme-li hodnotu výrazu, se kterou dynamický výraz pracuje, ihned se propočítá s novými hodnotami. Výraz můžeme udělat dynamický aplikujeme-li na něj funkci `Dynamic[]`. Uvedme příklad.

```
f[x_] := x  
Dynamic[Plot[f[x], {x, 0, 5}]]
```

Vykreslí se graf s přímkou úměrou. Změní-li se definice funkce na $f[x_] := x^2$, původní graf se automaticky překreslí na část paraboly.

3. Simulace chodu paprsku optickou soustavou

Naším cílem bylo vytvořit knihovnu funkcí, napsaných za účelem trasování paprsku optickou soustavou, a také interaktivní simulaci, použitelnou jak ve výuce na Přírodovědecké fakultě Univerzity Palackého, tak pro vysvětlení základů geometrické optiky na středních školách. Celý projekt je vytvořen ve Wolfram Mathematica 11.2.

V kapitole nejsou uvedeny celé kódy funkcí, jen jejich části. Celý kód naleznete na příloženém CD. Funkce použité v této kapitole jsou stručně popsány v kapitole 2.

3.1. Funkce vytvořené pro trasování paprsků optickou soustavou

Nejprve popíšeme vytvořené funkce, dohromady tvoří zmíněnou knihovnu. U jednotlivých funkcí nejsou ošetřeny vstupy. Pokud uživatel zadá místo čísla řetězec, funkce skončí chybou. Vstupy jsou ošetřeny až u samotné simulace, té se ovšem budeme věnovat později.

3.1.1. Reprezentace světelných paprsků

Každý paprsek je v našem modelu reprezentován přímkou a tu můžeme v případě dvojrozměrného prostoru jednoznačně popsat vztahem $y = ax + b$, proto jsme vytvořili dvě funkce, které ze dvou bodů nebo směrnice přímky a jednoho bodu vytvoří požadovaný tvar, resp. pravou stranu $ax + b$.

Věnujme se nejdříve funkci `SlopeInterceptForm[]`, která požaduje jako vstup směrnici přímky a dvoubodový list, který reprezentuje bod, kterým přímka prochází.

```
SlopeInterceptForm[sgm_, point_] := If[ $\dots$ ,  $\dots$ ,  $\dots$ ]
```

Je-li první souřadnice rovna `Infinity`, potom funkce vrátí konstantní funkci rovnu druhé souřadnici bodu, výstup tedy není závislý na směrnici přímky. Zobrazení bodu z nekonečna je vhodné k určení ohniska optické soustavy. Ač se zdá vrácená hodnota nesmyslná, v dalším se ukázala výhodnou, protože zobrazíme-li předmětové ohnisko zobrazí se do nekonečna. V případě, že první souřadnice není rovna `Infinity`, pak funkce `SlopeInterceptForm[]` vrátí výraz v již zmíněném tvaru $ax + b$.

Funkce `SlopeInterceptFormFromPoints[]` vrátí pravou stranu $ax + b$ v závislosti na dvou bodech. Oba jsou reprezentovány listem se dvěma prvky.

```
SlopeInterceptFormFromPoints[A_, B_] := Module[ $\dots$ , If[ $\dots$ ,  $\dots$ ,  $\dots$ ] ]
```

Ve funkci se nacházejí dvě podmínky srovnávající první souřadnici obou bodů s `Infinity` ze stejného důvodu jako u funkce `SlopeInterceptForm[]`. Jsou-li první souřadnice obou bodů různé od `Infinity`, je opět vrácen výraz ve tvaru $ax + b$.

3.1.2. Vytvoření lámavé plochy

V celém programu je každá sférická lámavá plocha představována jedním výrazem typu association list, který je vytvářen funkcí `CreateRefractingSurface[]`.

```
CreateRefractingSurface[r_, p_, n1_, n2_] :=  
< | "r" → r, "p" → p, "n1" → n1, "n2" → n2 | >
```

Funkce má čtyři vstupní parametry. První parametr r je poloměr lámavé plochy. Jeho znaménko musí respektovat znaménkovou konvenci 1.4.1. Druhým parametrem je p , tento parametr udává vzdálenost vrcholu lámavé plochy od počátku soustavy souřadné, může nabývat pouze kladných hodnot. Poslední dva parametry $n1$ a $n2$ udávají indexy lomu po řadě před a za lámavou plochou.

`CreateRefractingSurface[]` vrací association list, názvy klíčů pro poloměr, polohu a indexy lomu se shodují s názvy vstupních parametrů, ovšem je nutné je uvádět jako řetězec ("r").

3.1.3. Přidání ohniskových vzdáleností, optické mohutnosti a středu lámavé plochy

Association list vytvořený funkcí `CreateRefractingSurface[]` obsahuje plný popis sférické lámavé plochy, ovšem někdy se hodí znát i další vlastnosti lámavé plochy. Tyto vlastnosti doplní funkce `AddFeatures[]`.

```
AddFeatures[RS_] := Append[RS, <|...|>
```

Vstupem je association list představující lámavou plochu.³² Výstupem je stejný association list, ovšem s doplněnými klíči "f", "f1", "f2" a "S", které představují po řadě optickou mohutnost, předmětovou ohniskovou vzdálenost, obrazovou ohniskovou vzdálenost a vzdálenost středu lámavé plochy od počátku soustavy. První tři hodnoty jsme dopočetli dle rovností 28, 29 a 30, vzdálenost středu je zřejmá.

Jelikož může být poloměr lámavé plochy nekonečný, je ve funkci `AddFeatures[]` tento případ ošetřen pomocí funkce `Limit[]`.

3.1.4. Souřadnice průniku lámavé plochy a paprsku

Při tvorbě knihovny jsme zautomatizovali řadu geometrických úkolů. V tomto případě úlohu nalezení průsečíku lámavé plochy a paprsku. Funkce hned na začátku rozliší případ, kdy je poloměr lámavé plochy nekonečný a kdy ne. Je-li nekonečný, funkce určí průnik dvou přímek.

Není-li nekonečný, funkce řeší průnik kružnice³³ a přímky. Zde nastává další problém, kdy jako průnik dostáváme dva body. Z nich vybereme jeden dle znaménka poloměru lámavé plochy.

```
IntersectionOfRefractingSurfaceAndLine[SIF_, RS_] := Module[... ,  
  If[... , ... , ...]  
]
```

Vstupem jsou association list představující lámavou plochu s dopočtenými veličinami a výraz typu $ax + b$, který vrací funkce `SlopeInterceptForm[]` a `SlopeInterceptFormFromPoints[]`. Výstupem je uspořádaná dvojice čísel, dvouprvkový list, reprezentující souřadnice bodu průniku.

3.1.5. Výpočet souřadnic obrazů předmětových bodů po zobrazení přes konečný počet lámavých ploch

Dostáváme se k jedné ze stěžejních funkcí celé knihovny.

```
ThneedInvention[SoRS_, A_, sigma_] := Module[... ,  
  ...  
  Table[... , ...];  
  ...  
  <|"sgm" → sgm, "X1" → X1, "X2" → X2, "P" → P , "mez" → mez,  
  "Beta" → Beta|> ]
```

Vstupní parametry jsou `SoRS`³⁴, což je list složený z jednotlivých reprezentací lámavých ploch s přidávanými vlastnostmi,³⁵ a bod `A`, který chceme zobrazit přes optickou soustavu reprezentovanou

³²Výstup z `CreateRefractingSurface[]`.

³³To je křivka určená rovnicí $(x - s_1)^2 + (y - s_2)^2 = r^2$. Jelikož je střed kružnice v našem modelu vždy umístěn na optické ose, která splývá s osou x , vždy platí $s_2 = 0$.

³⁴System of Refracting Surface

³⁵Vrací jej funkce `AddFeatures[CreateRefractingSurface[...]]`

SoRS, spolu se směrnici *sigma* paprsku vycházejícího z kolmého průmětu bodu *A* do optické osy soustavy.

Funkce `ThneedInvention[]` vrací association list složený z obyčejných listů.³⁶ Rozeberme si nyní jednotlivé klíče:

- Klíč `"sgm"` ukazuje k listu, jehož počet prvků je o jedna větší, než je počet lámavých ploch v modelovaném systému. List obsahuje směrnice přímek vždy po průchodu lámavou plochou. Tedy na indexu jedna je směrnice před první lámavou plochou, na indexu dvě je směrnice za první lámavou plochou, na indexu tři je směrnice za druhou lámavou plochou atp.
- Klíč `"X1"` ukazuje k listu, jehož počet prvků je shodný s počtem lámavých ploch soustavy. Na *n*-tém indexu je vždy vzdálenost mezi vrcholem *n*-té lámavé plochy a zobrazovaným osovým bodem.
- Klíč `"X2"` ukazuje k listu se stejným počtem prvků jako v případě klíče `"X1"`. A opět na *n*-tém indexu je vždy vzdálenost mezi vrcholem *n*-té lámavé plochy a zobrazeným osovým bodem.
- Klíč `"P"` ukazuje k listu, který má opět stejný počet prvků jako *SoRS*, ovšem v tomto případě se skládá z bodů a ty jsou reprezentovány vždy dvouprvkovým listem.³⁷
- Pod klíčem `"mez"` se nachází souřadnice na ose *x*, ve kterých se mění směrnice paprsku.³⁸ List pod klíčem `"mez"` obsahuje o dva prvky více, než je počet lámavých ploch.
- Klíč `"Beta"` odkazuje na jednu hodnotu. Ta udává příčné zvětšení celé optické soustavy vzhledem k zobrazovanému předmětu.

Při konstrukci funkce jsme využili znalostí a vztahů zjištěných v odstavcích 1.4.3, 1.4.4 a 1.4.5. V principu jsme vzali předmětovou obrazovou vzdálenost a_1 , která odpovídá předmětovému bodu A_1 , jenž jsme chtěli zobrazit. Pomocí zobrazovací rovnice 25 jsme našli obrazový bod A'_1 a tedy i obrazovou vzdálenost a'_1 . Dle vztahu 17 jsme upravili obrazovou vzdálenost a'_1 na předmětovou vzdálenost a_2 a tím jsme našli i bod A_2 ,³⁹ který zobrazíme přes druhou lámavou plochu. Pokračujeme až dojdeme k poslední lámavé ploše.

To k plnému určení stopy paprsku nestačí. Musíme ještě určit průniky paprsků a lámavých ploch. Jelikož známe směrnici paprsku před první lámavou plochou a předmětový bod A_1 , dokážeme tento průnik P_1 určit. Jelikož platí $A_2 = A'_1$, pak přímka P_1A_2 je stopa paprsku mezi první a druhou lámavou plochou. Z této přímky již určíme jak směrnici stopy paprsku, tak bod průniku s druhou lámavou plochou P_2 .

Také jsme vždy ošetřili možnost nekonečného poloměru lámavé plochy pomocí podmínek `If[]` a funkce `Limit[]`. Pokud některá přímka, představující paprsek, neprotne příslušnou lámavou plochu, funkce skončí s chybou.

Funkce `ThneedInvention[]` nalezne potřebné údaje jen pro paprsky, které jsou různoběžné s optickou osou. Pro případ trasování paprsku rovnoběžného s optickou osou je nutno použít funkci `ThneedInventionForParallelRays[]`. Mimoběžné paprsky jsme neuvažovali vzhledem k tomu, že modelujeme jejich chování pouze ve dvojrozměrném prostoru.

3.1.6. Trasování paprsků rovnoběžných s optickou osou

Funkce `ThneedInventionForParallelRays[]` je v mnohém podobná funkci `ThneedInvention[]`. Má obdobné vstupy, ovšem bez směrnice přímky, ta je vždy rovna 0, i výstupy, s výjimkou počítání příčného zvětšení. Je určena k trasování paprsků procházejících mimosovým předmětovým bodem.

³⁶Až na klíč `"Beta"` který má pouze jeden prvek

³⁷Pro soustavu se dvěma lámavými plochami bude klíč `"P"` odkazovat na list typu $\{\{x_1, y_1\}, \{x_2, y_2\}\}$

³⁸Kromě souřadnic změn směrnice, obsahuje jako první a poslední hodnotu bod na optické ose, odkud se začíná paprsek vykreslovat a bod do kterého se vykresluje.

³⁹Platí $A_2 = A'_1$, ale neplatí $a'_1 = a_2$.

```

ThneedInventionForParallelRays[SoRS_, A_] := Module[... ,
  ...
  <|"sgm" → sgm, "X1" → X1, "X2" → X2, "P" → P , "mez" → mez,
  "A" → A|>
]

```

3.1.7. Výpočet osového zvětšení optickou soustavou

Zatímco příčné zvětšení je jednoznačně dáno jedním předmětovým osovým bodem a optickou soustavou, k určení osového zvětšení je třeba dvou osových předmětových bodů A_1 a B_1 , jelikož zvětšení γ je mírou zvětšení vzdáleností $|A_1B_1|$ a $|A'_jB'_j|$, proto jej nepočítáme ve funkci `ThneedInvention[]`.

```

AxisMagnification[SoRS_, Ray1_, Ray2_] := ...

```

Osové zvětšení se dá snadno určit ze vztahu 35, proto si `AxisMagnification[]` bere za vstup `SoRS`⁴⁰ a dva paprsky `Ray1` a `Ray2`, které jsou představovány výstupy z funkce `ThneedInvention[]`.⁴¹ Výstupem je číslo udávající osové zvětšení.

3.1.8. Hlavní body a ohniska optické soustavy

Poslední dvě funkce, které se týkají stopování paprsků optickou soustavou, slouží k určení ohnisek, ohniskových vzdáleností, hlavních bodů a hlavních vzdáleností.

K určení předmětové ohniskové vzdálenosti dle odstavce 1.4.6 je potřeba stopovat paprsek rovnoběžný s optickou osou obrácenou soustavou. Funkce `ReverseSystemOfRefractingSurface[]` soustavu obrátí. Poslední lámavá plocha je umístěna na místo první, znaménka poloměrů se změní, index lomu n_i se zamění za index lomu n'_i . Funkce si bere jako vstup `SoRS` a vrací objekt stejného typu.

```

ReverseSystemOfRefractingSurface[SoRS_] := Module[... , ...]

```

Funkce `PropertiesOf[]` již počítá ohniska, ohniskové vzdálenosti i hlavní vzdálenosti. `PropertiesOf[]` potřebuje jako vstup `SoRS` a vrací association list s klíči "f1", "h1", "F1" a "f2", "h2", "F2". Klíče odpovídají ohniskovým vzdálenostem, hlavním vzdálenostem a ohniskům.

```

PropertiesOf[SoRS_] := Module[... ,
  ...
  <|"f1" → f1, "h1" → h1, "F1" → h1 + f1, "f2" → f2, "h2" → h2,
  "F2" → h2 + f2|> ]

```

Kód funkce `PropertiesOf[]` můžeme rozdělit na dvě téměř identické části. V první hledáme obrazové ohnisko a obrazovou ohniskovou vzdálenost zadané optické soustavy. V druhé části hledáme opět obrazové ohnisko a obrazovou ohniskovou vzdálenost, tentokrát již obrácené soustavy. Takové obrazové ohnisko má význam předmětového ohniska pro zadanou soustavu.

⁴⁰System of Refracting Surfaces, který obsahuje informace o indexu lomu.

⁴¹Struktury obsahují informaci o příčném zvětšení.

Pro určení ohniskové vzdálenosti f' jsme užili vztahu 31. V případě, že byla některá vzdálenost z 31 nekonečná, dosadili jsme místo ní číslo 1. Obrazovou hlavní vzdálenost h' jsme určili dle odstavce 1.4.6.

Tímto máme popsány všechny části knihovny určené k trasování paprsku. V další podkapitole se budeme věnovat pouze tvorbě dynamického modelu a vizualizaci dosažených výsledků.

3.2. Vizualizace dosažených výsledků

V této části se budeme zabývat grafickou reprezentací paprsků a optických soustav.

3.2.1. Vykreslování paprsků

K vykreslování paprsků jsme vytvořili funkce `ShowRay[]`, která vrátí list složený z `graphics`, které jsou vytvořeny funkcí `Plot[]`. Skládáním jednotlivých úseček se budeme zabývat později. Parametry funkce jsou:

- `Ray` je reprezentace paprsku vrácená funkcí `ThneedInvention[]`;
- `SoRS` reprezentuje optickou soustavu, jedná se o list složený z více association list, které představují jednotlivé lámavé plochy.

`ShowRay[]` má dva volitelné parametry. Prvním je `"RayColor"`, který mění barvu paprsku. A druhým je `"LinkFocusAndLastIntersection"`, který spojí, pokud je nastaven na `True`, obrazové ohnisko s průsečíkem paprsku a poslední lámavé plochy.

Způsob vykreslení paprsku před první lámavou plochou se liší od způsobu vykreslení paprsku za první lámavou plochou, tj. před druhou, třetí, atp. V prvním případě se používá směrnice paprsku a předmětového bodu, dále už směrnice a obrazového bodu. Meze vykreslení jsou určeny ve funkci `ThneedInvention[]`.

```
Options[ShowRay]={"RayColor" → Black,  
  "LinkFocusAndLastIntersection" → False}  
ShowRay[Ray_, SoRS_, OptionPattern[]]:=Module[... ,...]
```

3.2.2. Vizualizace předmětů a obrazů

`ShowObject[]` vrátí list s prvky `Graphics`, které vizualizují předmět a obraz předmětu. Parametry funkce jsou:

- `point` je dvouprvkový list představující vrchol předmětu;
- `Ray` je reprezentace paprsku vrácená funkcí `ThneedInvention[]`;
- `SoRS` reprezentuje optickou soustavu, jedná se o list složený z více association list, které představují jednotlivé lámavé plochy.

Pomocí volitelného parametru `"ObjectColor"` můžeme změnit barvu předmětů a obrazů, což zlepšuje přehlednost při zobrazení více předmětů.

Součástí listu jsou i reprezentace speciálních úseček, které se zobrazí v případech:

1. předmět se nachází za první lámavou plochou, doplní se úsečka;⁴²
2. chceme zobrazit pouze rovnoběžné paprsky, v tomto případě je nutné nastavit parametr `"ShowRay"` na `False`.

⁴²Vhodné pro ukázkou zvětšení předmětu v hlavním bodě.

```
Options[ShowObject] = {"ParallelRays" → {<|"P"->0,0|>}, "ShowRay" → True,
  "ObjectColor" → Red}
ShowObject[point_, Ray_, SoRS_, OptionsPattern[]] := Sequence[
  If[⋯, ⋯, ⋯], If[⋯, ⋯, ⋯], If[⋯, ⋯, ⋯]
]
```

Kód funkce `ShowObject`⁴³ je parametrem funkce `Sequence`, protože vrací hodnoty, které jsou vždy použity jako parametr funkce `Show`. Kdyby funkce `ShowObject` vrátila list, pak by evaluace funkce `Show[ShowObject[], ⋯]` skončila chybou: "Could not combine the graphical object...".

3.2.3. Určení mezí pro vykreslovanou oblast

Pomocí funkcí `LimitX` a `LimitY` určíme omezení pro oblast vykreslování. Je-li vzdálenost mezi předmětem a obrazem mnohem větší, než vzdálenost mezi první a poslední lámavou plochou, je někdy výhodné pozorovat pouze průchod optickou soustavou. Obě funkce vrací asociation list obsahující klíče "min" a "max". Jak názvy funkcí a klíčů napovídají, jedná se vždy o mezní hodnoty zobrazení v osách x a y.

Parametry funkce `LimitX` jsou:

- *SoRS* reprezentuje optickou soustavu, jedná se o list složený z více association list, které představují jednotlivé lámavé plochy;
- *Prop* je asociation list složený z vlastností soustavy, je vrácen funkcí `PropertiesOf`;
- *Rays* je list složený z reprezentací jednotlivých paprsků.

Funkce má dva další volitelné parametry "ShowFocusToFocus" a "ShowSystemOnly". Nastavíme-li jeden z nich na `True`, změní se rozsah zobrazení v ose x, z módu "zobrazit vše" na zvolený režim. Výchozí hodnoty obou parametrů jsou `False`.

```
Options[LimitX] = {"ShowFocusToFocus" → False, "ShowSystemOnly" → False}
LimitX[SoRS_, Prop_, Rays_, OptionsPattern[]] := Module[⋯, ⋯]
```

Funkce `LimitX` vezme obě ohniska, hlavní body, vrchol první a poslední lámavé plochy, pozice předmětů a obrazů a vybere z nich dva nejvzdálenější, pokud není zvolen speciální mód.

Parametry funkce `LimitY` jsou:

- *Rays* je list složený z reprezentací jednotlivých paprsků;
- *ParallelRays* je list složený z reprezentací jednotlivých rovnoběžných paprsků;
- *points* je list složený z listů, které reprezentují souřadnice předmětů.

`LimitY` má jeden volitelný parametr "ObjectChoose". Je-li nastaven na `False`, nezahrnou se do postupu určení krajních hodnot pozice předmětů. Výchozí hodnota je `True`.

Funkce `LimitY` funguje obdobně jako funkce `LimitX`.

```
Options[LimitY] = {"ObjectChoose" → True}
LimitY[Rays_, ParallelRays_, points_, OptionsPattern[]] := Module[⋯, ⋯]
```

⁴³A dále i kód funkce `ShowLenses`.

3.2.4. Grafická reprezentace lámavých ploch a význačných bodů

Pro vykreslení lámavých ploch a význačných bodů, jako jsou ohniska nebo hlavní body, jsme vytvořili funkci `ShowLenses[]`. Parametry funkce jsou:

- `SoRS` reprezentuje optickou soustavu, jedná se o list složený z více association list, které představují jednotlivé lámavé plochy;
- `ym` je dvouprvkový association list s klíči "min" a "max", ten je vrácen funkcí `LimitY[]`;
- `Prop` je association list složený z vlastností soustavy, je vrácen funkcí `PropertiesOf[]`.

Je-li jediný volitelný parametr "ShowPoints" nastaven na `True`, vykreslí se kromě lámavých ploch i ohniska a hlavní body. V opačném případě se nevykreslí. Výchozí hodnota parametru je `True`.

K vykreslování lámavých ploch využíváme funkci `ContourPlot[]`. Dle znaménka a konečnosti poloměru lámavých ploch se liší způsob vykreslení.

```
Options[ShowLenses] = {"ShowPoints" -> True}
ShowLenses[SoRS_, ym_, Prop_, OptionsPattern[]] := Sequence[
  Table[...], If[...], ...]
]
```

3.3. Tvorba dynamického modelu

V následující části se budeme věnovat využití funkce `Manipulate[]` pro tvorbu dynamického modelu a budeme řešit problematiku ošetření vstupů.

3.3.1. Kontrola zadání nekonečna

Prostředí Wolfram Mathematica nabízí funkci `Restricted[]`, která omezuje vstupy do formuláře.⁴⁴ Bohužel se nám nepodařilo ji úspěšně použít k omezení vstupů na reálná čísla z intervalu $< -1000; 1000 >$ spolu s nevlastní hodnotou `Infinity`.

Proto jsme vytvořili funkci `InfinityControl[]`, která v případě porušení nastavených parametrů vrátí obdobnou hodnotu jako funkce `Restricted[]`.

```
WrongValue = Failure["InterpretationFailure",...];
InfinityControl[x_] := Module[... ,
  If[...], ...]
]
```

3.3.2. Tvorba rozhraní a volba počtu čoček

Při tvorbě formuláře jsme měli na mysli hlavně jednoduchost. Jestliže bude simulace využívána při výuce, případný student by neměl být zmatený z formuláře, ale měl by sledovat změny v chodu paprsku optickou soustavou. Obětí za to je ztráta některých funkcí v dynamickém modelu, ovšem případní zájemci, kteří umějí prostředí Mathematica využívat, mohou použít knihovnu k tvorbě složitějších modelů.

Ve formuláři jsme se omezili na zobrazení maximálně čtyřmi čočkami a zobrazení maximálně dvou předmětů. Ke každému předmětu přísluší jeden paprsek rovnoběžný s optickou osou a jeden vycházející z kolmé projekce předmětu do optické osy pod zadaným úhlem.

⁴⁴Například může omezit vstup pouze na celá čísla z intervalu $< -10; 10 >$, popřípadě omezí vstup pouze na textové řetězce.

Samotnou dynamiku našeho modelu obstará funkce `Manipulate[]`. Jelikož má změna počtu lámavých ploch v optické soustavě vliv na počet vstupních polí potřebných k jejímu jednoznačnému určení, nepodaří se nám vytvořit dynamický model pouze s použitím jedné funkce `Manipulate[]`, ale museli jsme do původní funkce `Manipulate[]` zanořit další.

Z toho důvodu jsme vytvořili čtyři funkce `OneLens`, `TwoLenses`, `ThreeLenses` a `FourLenses` volající funkci `Manipulate[]`.

```
OneLens := Manipulate[... , ...]
```

Každá z těchto funkcí zobrazí rozhraní s výchozími hodnotami a model spočtený pro výchozí hodnoty. Mezi těmito funkcemi přepíná námi vytvořená funkce `Interface`.

```
Interface := Manipulate[... , ...]
```

Zabývejme se funkcí `OneLens`.⁴⁵ Celá je obalená do funkce `Manipulate[]`, ta má dva vstupní parametry. Jedním je prováděný kód, ve druhé se definuje změna proměnných použitých v kódu. V části prováděného kódu nejprve vytváříme reprezentace zobrazovaných bodů a optických soustav složených z lámavých ploch pomocí `CreateRefractingSurface[]` a `AddFeature[]`. Poté již počítáme průběhy paprsků optickou soustavou funkcemi `ThneedInvention[]` a její alternativou pro rovnoběžný paprsek, nakonec pomocí funkce `Show[]` spojíme grafické prvky vytvořené funkcemi `ShowRay[]`, `ShowObject[]` a `ShowLenses[]`.

V části definic proměnných vytváříme samotný formulář funkcemi `Control[]` a `FormObject[]`, ale zároveň vypisujeme vybrané spočtené hodnoty, například ohniska, hlavní body, zvětšení atp. Jelikož počítáme vlastnosti optické soustavy a paprsků z hodnot, které může uživatel měnit, musíme vypisované proměnné obalit do funkce `Dynamic[]`. V opačném případě se buď zobrazí neaktuální, popř. falešné, hodnoty nebo celá simulace skončí chybou. Formulář formátujeme pomocí funkce `Grid[]`, která zarovnává jednotlivé prvky do mřížky.

3.4. Návod k použití dynamického rozhraní

Pokud zavoláme funkci `Interface`, zobrazí se nám vytvořená simulace, tu můžete vidět na obrázku 15. Vidíme dvě do sebe vnořená okna.⁴⁶ V levém horním rohu můžeme zvolit počet čoček. Dle naší volby se rozhraní, tedy vnořené okno, upraví.

Prostor vnořenému rozhraní můžeme rozdělit na tři části:

- prostor vstupních hodnot, vlevo nahoře;
- prostor výpisu spočtených hodnot, vpravo nahoře;
- grafická reprezentace optické soustavy, dole.

3.4.1. Parametry předmětů a paprsků

Změní-li se parametry soustavy nebo paprsků a stiskneme-li tlačítko "Vizualizuj!", aktualizují se zbývající dvě části. Prvních šest políček formuláře je věnovaných nastavení parametrů paprsků a předmětů. Počátek optické soustavy, tj. vrchol první lámavé plochy, se automaticky umísťuje do počátku soustavy souřadné. Bude-li pozice předmětu -10 bude umístěn 10 jednotek před první lámavou plochu. To je z důvodu zachování znaménkové konvence popsané v odstavci 1.4.1. V simulaci není umožněno nastavení nezáporné pozice předmětu, proto může být umístěn jen před první lámavou plochu. Měří-li se výška předmětu nahoru od optické osy, je kladná.

Při volbě výšky předmětu i volbě velikosti úhlu pod kterým se paprsek šíří, musíme mít na paměti, že pracujeme v paraxiálním prostoru. Zvolíme-li úhel 10°, výšku předmětu dvě jednotky a poloměr lámavé plochy 3 jednotky, už se v paraxiálním prostoru nenacházíme. Také musíme

⁴⁵Funkce `TwoLenses`, `ThreeLenses` a `FourLenses` jsou obdobné, proto se jimi nebudeme více zabývat. Jediným rozdílem je zvětšení počtu vstupních polí formuláře a více lámavých ploch.

⁴⁶Důsledek použití zanořených funkcí `Manipulate[]`.

mít na paměti, že se paprsek a lámavá plocha nemusí při špatné volbě vzdáleností a úhlů protnout.⁴⁷ Bohužel v programu není tato chyba ošetřena a v oblasti grafické reprezentace soustavy se objeví nesmyslné výpisy.

Umístíme-li předmět do blízkosti předmětového ohniska, zobrazí se do velké vzdálenosti od soustavy, v tom případě je možné, že se celá soustava zobrazí do relativně malé oblasti a nebude možné rozeznat jednotlivé části.

Ve výchozím nastavení se druhý předmět nezobrazuje, protože je v políčku "Pozice 2. předmětu" zadána nula. Nastaví-li se pozice tohoto předmětu, zobrazí se.

3.4.2. Parametry optické soustavy

U optické soustavy můžeme nastavovat indexy lomu prostředí, ve kterém se paprsek šíří. Tento index lomu musí být u dvou sousedních prostředí různý, jinak dojde k chybě dělení nulou. Z fyzikálního hlediska k takovému lomu nedochází. Znaménko poloměru lámavé plochy se řídí znaménkovou konvencí. Má-li docházet k lomu na vnitřní straně části kružnice, poloměr musí být záporný a obráceně.

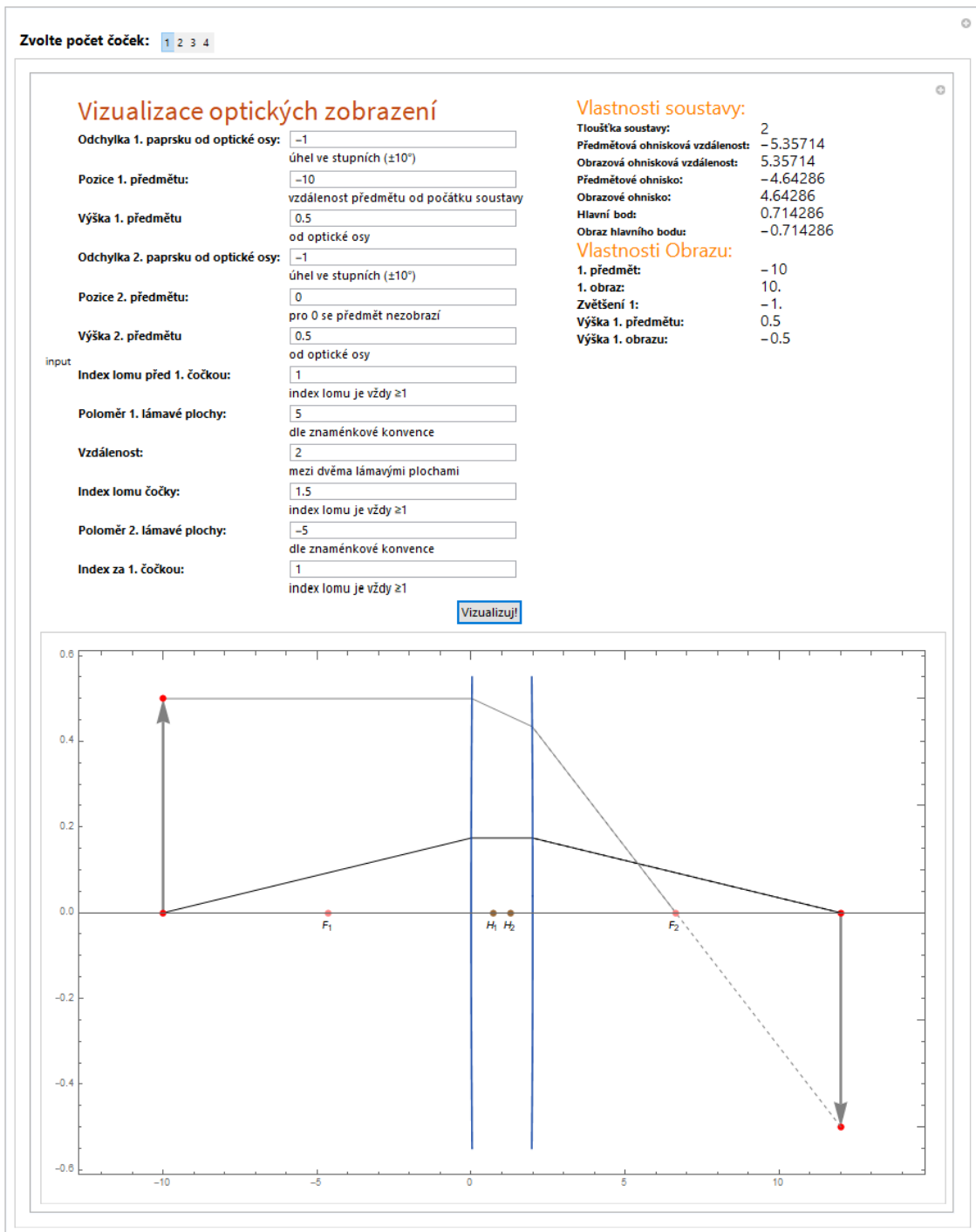
3.4.3. Vlastnosti optické soustavy

V prostoru výpisu spočtených hodnot nacházíme celkovou tloušťku soustavy, ohniskové vzdálenosti, polohy ohnisek a hlavních bodů. Těchto informací není mnoho, ovšem je snadné rozšířit nebo změnit počet vypisovaných parametrů a pokud by o to případný uživatel stál může tuto změnu provést.

3.4.4. Vlastnosti předmětů a obrazů

V této části se vypisují pozice předmětů, obrazů, také jejich výšky a příčné zvětšení, popřípadě osové zvětšení.

⁴⁷Může dojít i k situaci, kdy paprsek protne nejprve druhou lámavou plochu a až pak první, v tom případě simulace vytvoří graf, ovšem bude nesmyslný.



Obrázek 15 | Vytvořená simulace.

3.5. Ukázky grafických výstupů

Modelujme nejprve průchod paprsku jednou lámavou plochou. U tohoto příkladu si uvedeme i kód, v dalších příkladech tak již činit nebudeme. Celý kód je k dispozici v příloze. Nejprve zdefinujeme proměnné určující předmět, paprsek a optickou soustavu.

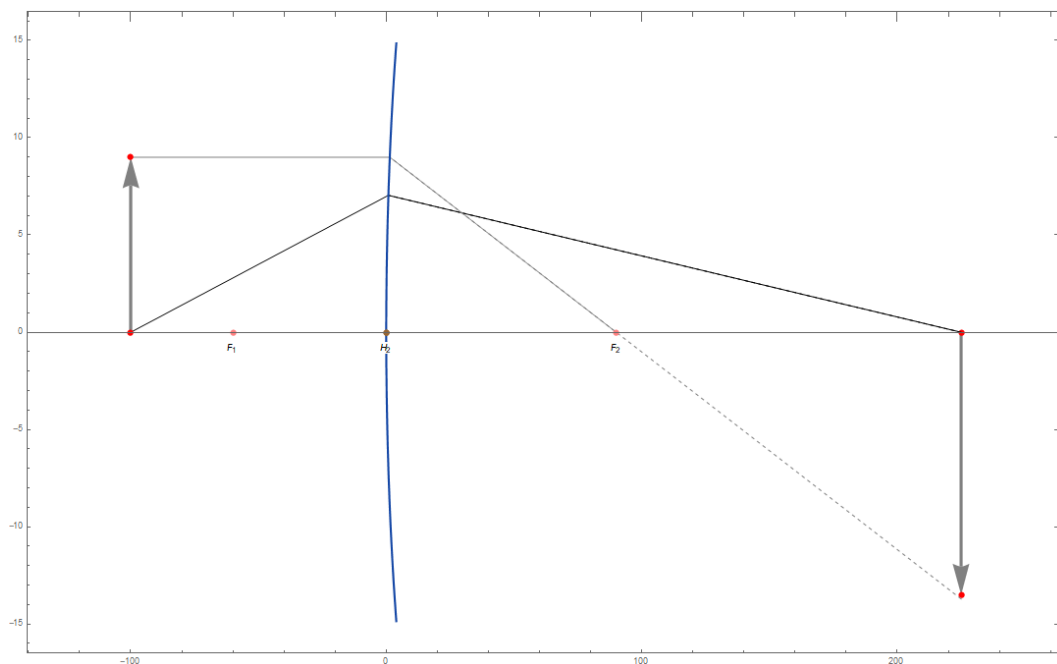
```
(*Zadání indexů lomu, poloměru, polohy objektu*)
ns = {1,1.5}; ds = {}; rs = {30}; Objs = {{-100,9}};
Origin = 0;      (*pozice první lámavé plochy*)
T = {Origin + Objs[[1]][[1]], Objs[[1]][[2]]};
tan1 = -4 Pi / 180 //N;      (*úhel šíření paprsku*)
(*Popis optické soustavy*)
ds = Prepend[ds, Origin];
Table[ds[[i]] = ds[[i-1]] + ds[[i]], i,2,Length[rs]];
SystemOfRS = Table[
  AddFeatures[CreateRefractingSurface[rs[[i]], ds[[i]], ns[[i]], ns[[i+1]]],
  {i, 1, Length[rs]}
]
```

Dále spočítáme ohniska a hlavní body pomocí `PropertiesOf[]` a vytvoříme reprezentaci paprsků, také určíme ohraničení vykreslované oblasti.

```
Prope = PropertiesOf[SystemOfRS];      (*Vlastnosti optické soustavy*)
a = ThneedInvention[SystemOfRS, T, tan1];      (*Paprsky*)
c = ThneedInventionForParallelRays[SystemOfRS, T];      (*Rovnoněžné paprsky*)
ym = [{a}, {c}, {T}];      (*Ohraničení vykreslení*)
xm = LimitX[SystemOfRS, Prope, {a}];
```

A konečně použijeme funkci `Show` k zobrazení našeho paprsku a soustavy. Výstup je zobrazen na obrázku 16.

```
Show[
  ShowRay[a, SystemOfRS],
  ShowRay[c, SystemOfRS, "RayColor"->Gray],
  ShowObject[T, a, SystemOfRS, "ParallelRays"->{c}, "ObjectColor"->Red],
  ShowLenses[SystemOfRS, ym, Prope],
  PlotRange->{{xm["min"], xm["max"]},{ym["min"], ym["max"]}},
  AxesOrigin->{xm["min"], 0},
  Frame->{True, True, True, True},
  ImageSize->1200 ]
```

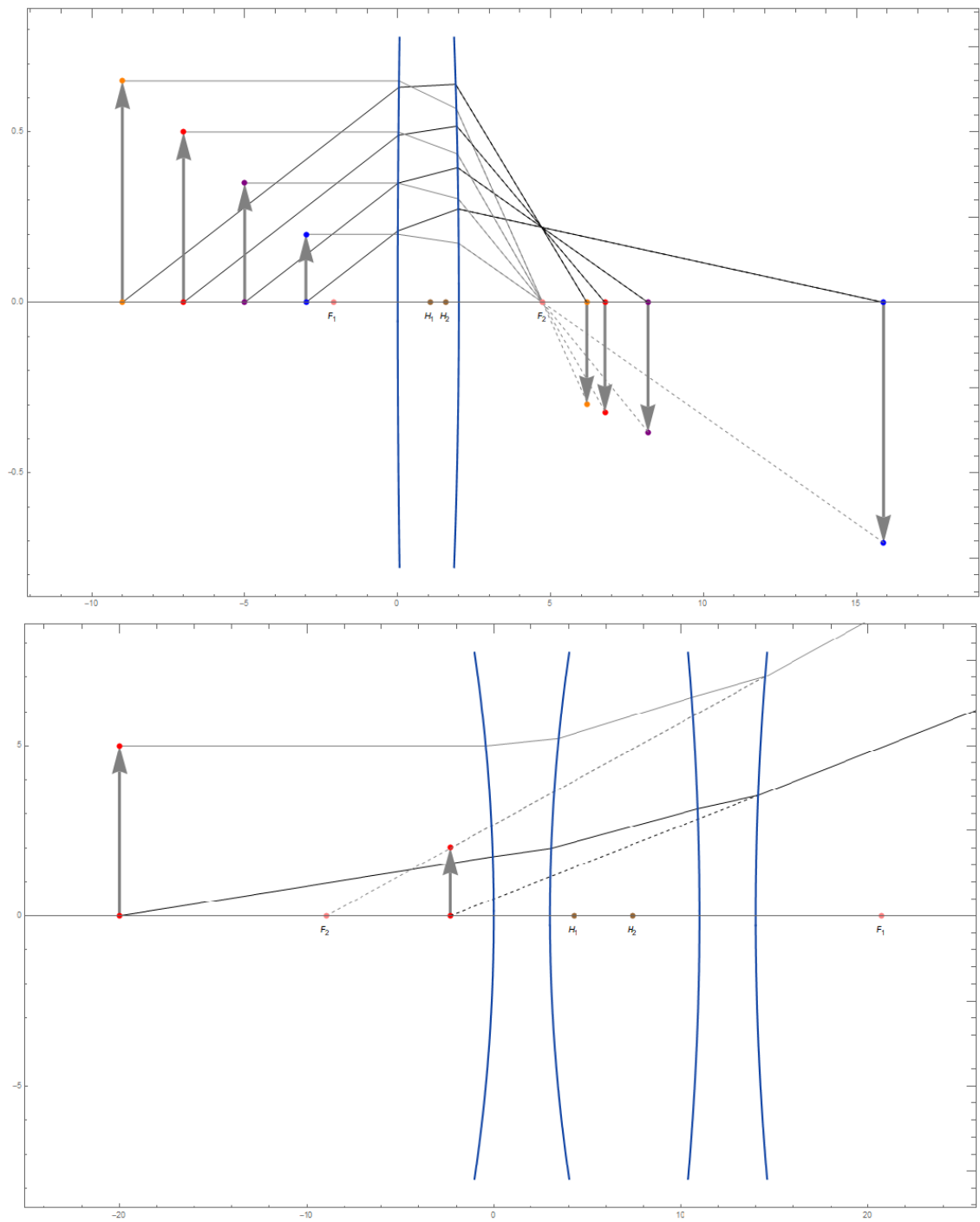


Obrázek 16 | Vizualizace paprsku procházejícího jednou lámavou plochou.

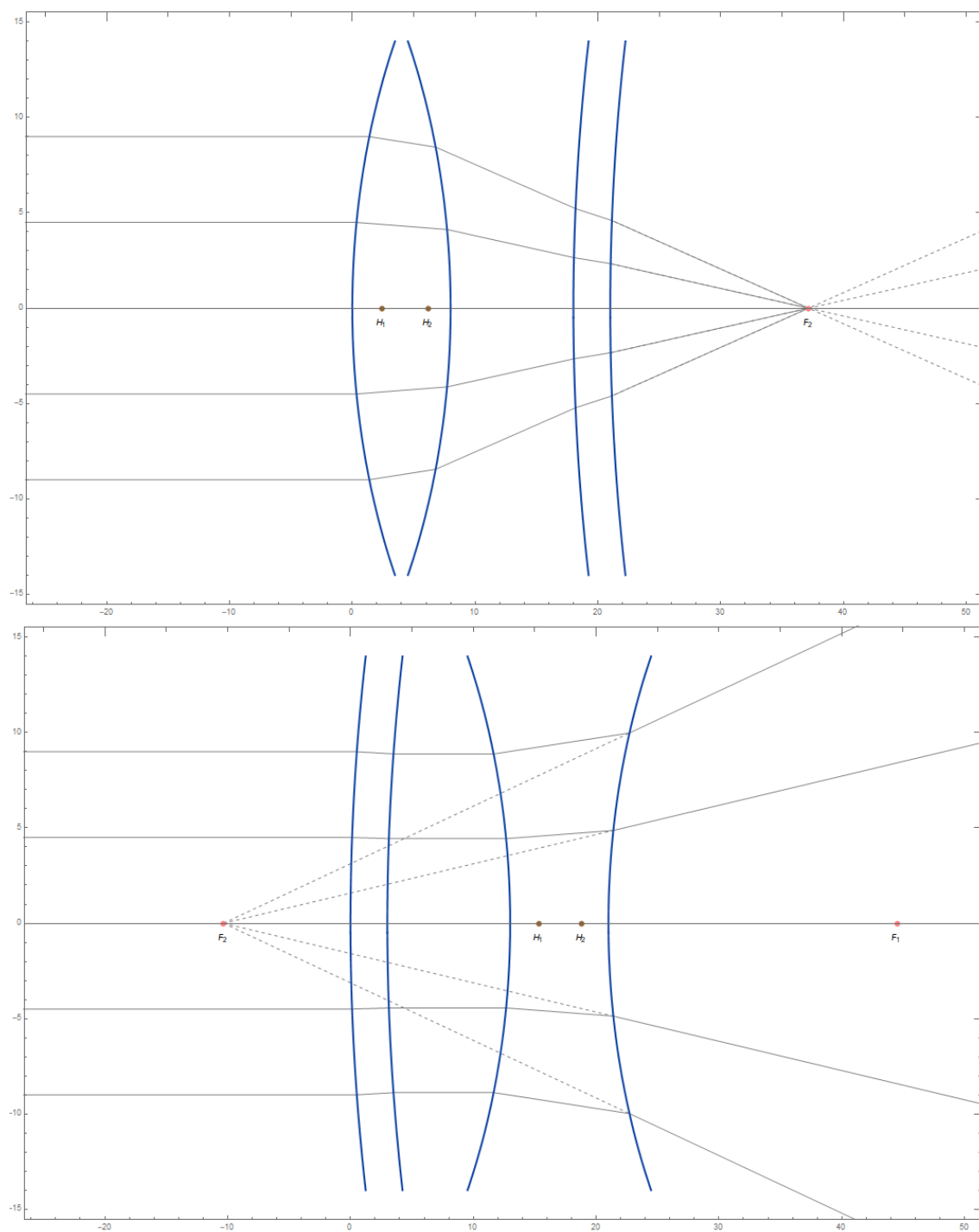
Nakonec necháme prostředí Wolfram Mathematica vypsat vybraná spočtená data zavoláním `Prope`. A dostaneme výsledek.

```
<|"f1" → -60., "h1" → 0., "F1" → -60., "f2" → 90., "h2" → 0.,  
"F2" → 90. |>
```

Jak již jsme řekli, možnosti knihovny jsou mnohem větší, než možnosti simulace, jak můžete vidět na obrázcích 17 a 18.



Obrázek 17 | *Nahoře: zobrazení čtyř předmětů skrz jednu tlustou čočku; čím více se předmět blíží k předmětovému ohnisku, tím se obraz vzdaluje od optické soustavy. Dole: zobrazení neskutečného obrazu dvěma tlustými čočkami.*



Obrázek 18 | Chod rovnoběžných paprsků přes optickou soustavu složenou ze dvou tlustých čoček.

3.6. Optická lavice

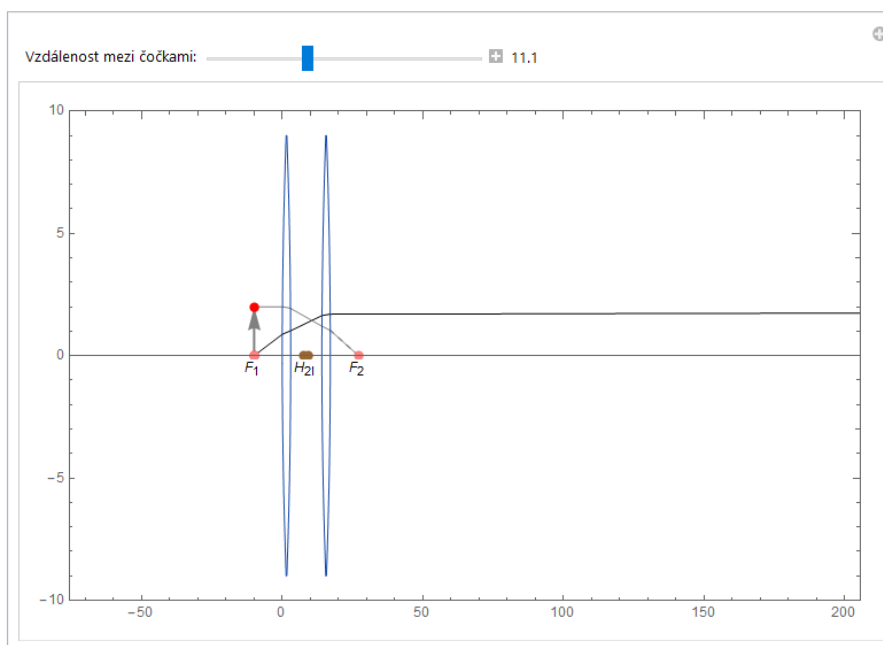
Na závěr kapitoly uvedme ještě jedno využití knihovny pro dynamické modelování optické soustavy. Budeme modelovat optickou lavici, viz obr. 19, z Pevnosti Poznání.⁴⁸ Jedná se o dvě tlusté spojné čočky, přičemž s jednou z nich je možno pohybovat a tím měnit vlastnosti soustavy.



Obrázek 19 | Exponát optická lavice z expozice Světlo a tma, interaktivního muzea vědy Pevnost poznání.

Bohužel se nám nepodařilo dohledat poloměry a index lomu čoček. Poloměry a index lomu v modelu tedy nemusejí odpovídat skutečnosti, ale v tomto modelu jde spíše o vizualizaci změny chodu paprsku soustavou, která je způsobená proměnlivou vzdáleností čoček, než o přesné výsledky.

Obdobně jako u jednoho z příkladů z podkapitoly 3.5 jsme vizualizovali chod paprsku soustavou dvou čoček. V tomto případě jsme celé kód obalili do funkce `Manipulate[]` a jako dynamickou proměnnou nastavili vzdálenost mezi čočkami. Výsledkem je vizualizace, viz obr. 20, kde pomocí slideru měníme vzdálenost čoček.⁴⁹



Obrázek 20 | Dynamický model optické lavice.

⁴⁸Interaktivní muzeum vědy Univerzity Palackého v Olomouci, <https://www.pevnostpoznani.cz/>.

⁴⁹Bohužel při běhu dynamického modelu na našem hardwaru dochází k opožděným reakcím na změnu.

4. Příklady řešené pomocí simulace a přímým výpočtem

V této kapitole budeme řešit tři typické příklady. Úlohou bude určit zvětšení, ohniskové vzdálenosti atp. Kompletní kód k příkladům naleznete na příložením CD.

4.1. 1. příklad

Začněme příkladem z [7], str. 71:

“Určete polohy ohnisek, ohniskové vzdálenosti a polohy hlavních bodů u soustavy:

$$\begin{aligned}r_1 &= 30, & n_1 &= 1; \\r_2 &= -60, & d_1 &= 6, & n'_1 &= n_2 = 1,5; \\r_3 &= 20, & d_2 &= 10, & n'_2 &= n_3 = 1; \\r_4 &= 40, & d_3 &= 6, & n'_3 &= n_4 = 1,5; \\& & & & n'_4 &= 1''.\end{aligned}$$

Abychom určili obrazové ohnisko, ohniskovou vzdálenost a hlavní bod, budeme trasovat soustavou paprsek rovnoběžný s optickou osou. Předmětová vzdálenost předmětu je $a_1 = -\infty$. Pro získání dílčích předmětových a obrazových vzdáleností užijeme vztah 31. Dle zobrazovací rovnice 25 a vztahu 17, platí:⁵⁰

$$\begin{aligned}a_1 &= -\infty, & a'_1 &= \lim_{a_1 \rightarrow -\infty} \frac{n'_1}{\frac{n_1}{a_1} + \frac{n'_1 - n_1}{r_1}} = 90; \\a_2 &= a'_1 - d_1 = 84, & a'_2 &= \frac{n'_2}{\frac{n_2}{a_2} + \frac{n'_2 - n_2}{r_2}} = 38,18; \\a_3 &= a'_2 - d_2 = 28,18, & a'_3 &= \frac{n'_3}{\frac{n_3}{a_3} + \frac{n'_3 - n_3}{r_3}} = 24,8; \\a_4 &= a'_3 - d_3 = 18,8, & a'_4 &= \frac{n'_4}{\frac{n_4}{a_4} + \frac{n'_4 - n_4}{r_4}} = 14,86.\end{aligned}$$

Dle vztahu 31, pak určíme ohniskovou vzdálenost jako

$$f' = \frac{a'_1 a'_2 a'_3 a'_4}{a_2 a_3 a_4} = 28,4585.$$

Obrazovou hlavní vzdálenost určíme jako $h' = x'_4 - f' = -13,5969$.

Předmětovou ohniskovou a hlavní vzdálenost určíme z chodu rovnoběžného paprsku obrácenou soustavou. Opět hledáme obrazovou ohniskovou a hlavní vzdálenost. Popíšme si otočenou optickou soustavu:

$$\begin{aligned}r_1 &= -40, & & n_1 &= 1; \\r_2 &= -20, & d_1 &= 6, & n'_1 &= n_2 = 1,5; \\r_3 &= 60, & d_2 &= 10, & n'_2 &= n_3 = 1; \\r_4 &= -30, & d_3 &= 6, & n'_3 &= n_4 = 1,5; \\& & & & n'_4 &= 1.\end{aligned}$$

⁵⁰Pro $a = -\infty$, užijeme limitu, stejně tak pro případný nekonečný poloměr lámavé plochy.

Určíme předmětové a obrazové vzdálenosti zobrazovaného bodu:

$$\begin{array}{ll} a_1 = -\infty, & a'_1 = -120; \\ a_2 = -126, & a'_2 = 76, 36; \\ a_3 = 66, 36, & a'_3 = 64, 1; \\ a_4 = 58, 1, & a'_4 = 23, 54. \end{array}$$

Ohnisková vzdálenost je pak $f = -28, 46$ a hlavní vzdálenost $h = 4, 92$.⁵¹

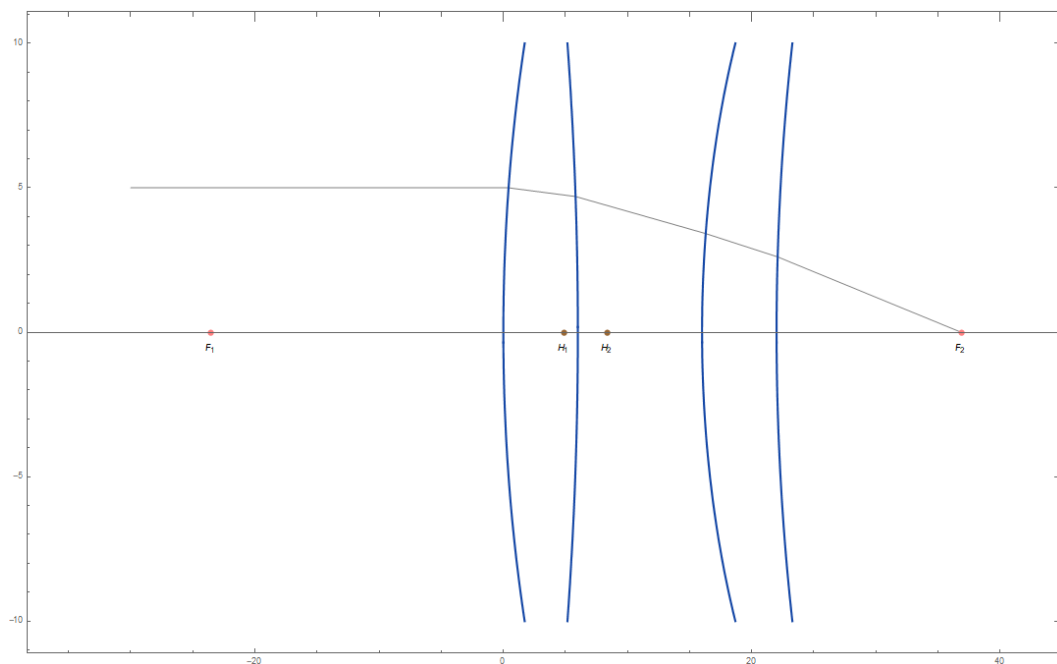
Spočtíme nyní stejný příklad pomocí naší knihovny. Zadejme soustavu do prostředí Mathematica a zavolejme funkci `PropertiesOf[...]`. Dostaneme.

```
<|"f1" → -28.4585, "h1" → 4.92095, "F1" → -23.5375, "f2" → 28.4585,
  "h2" → -13.5968, "F2" → 36.8617|>
```

K mezivýsledkům se dostaneme zavoláním funkce `ThneedInventionForParallerRays[...]`.

```
<|..., "X1" → {-∞, 84., 28.1818, 18.8},
  "X2" → {90., 38.1818, 24.8, 14.8617}, ... |>
<|..., "X1" → {-∞, -126., 66.3636, 58.0976},
  "X2" → {-120., 76.3636, 64.0976, 23.5375}, ... |>
```

Navíc můžeme zobrazit rovnoběžný paprsek procházející optickou soustavou, viz obr. 21.



Obrázek 21 | *Rovnoběžný paprsek procházející optickou soustavou z 4.1.*

⁵¹Vypočítáme-li obrazovou ohniskovou vzdálenost opačné soustavy dle 31, vyjde nám kladně. Ovšem my počítáme předmětovou ohniskovou vzdálenost zadané soustavy, proto "-".

4.2. 2. příklad

Určete příčné a osové zvětšení předmětů A[-4;3] a B[-20;3] optickou soustavou:

$$\begin{array}{lll}
 r_1 = 40, & & n_1 = 1; \\
 r_2 = -40, & d_1 = 6, & n'_1 = n_2 = 1,62; \\
 r_3 = 40, & d_2 = 10, & n'_2 = n_3 = 1; \\
 r_4 = -40, & d_3 = 6, & n'_3 = n_4 = 1,53; \\
 & & n'_4 = 1.
 \end{array}$$

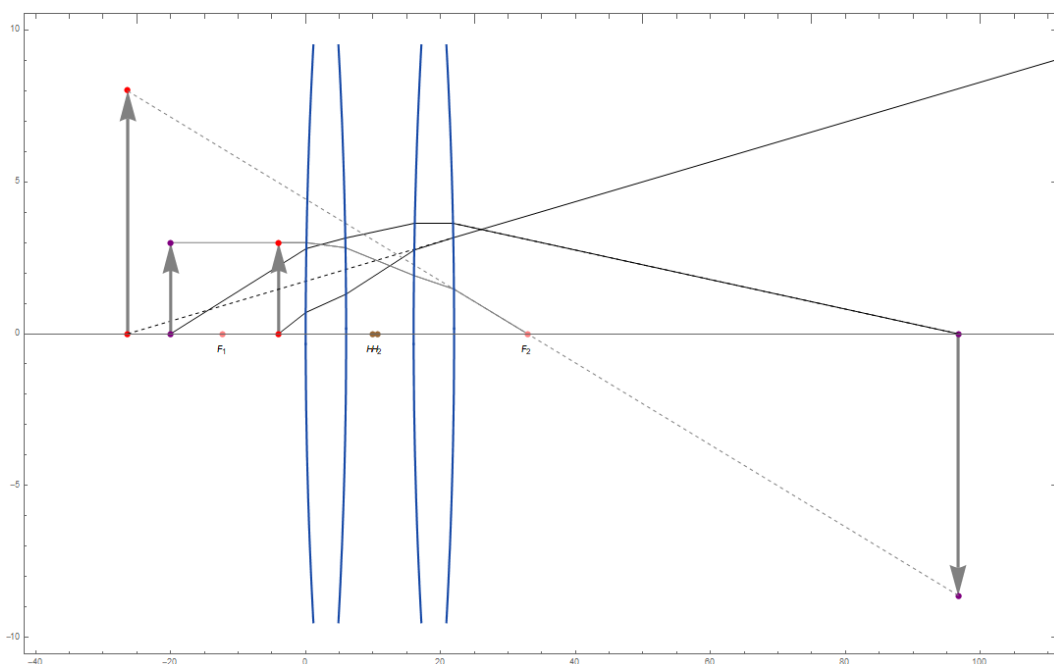
Osové zvětšení určíme z obou příčných podle vztahu 35 a ty dle 34. Potřebujeme znát dílčí předmětové a obrazové vzdálenosti. Určíme je obdobně jako v předchozím příkladu, začneme bodem A:

$$\begin{array}{ll}
 a_1 = -4, & a'_1 = -6,91; \\
 a_2 = -12,91, & a'_2 = -9,09; \\
 a_3 = -19,09, & a'_3 = -39,1; \\
 a_4 = -45,1, & a'_4 = -48,37.
 \end{array}$$

Pro příčné zvětšení dostaneme $\beta_A = 2,67$. Postupujme obdobně pro předmět B:

$$\begin{array}{ll}
 b_1 = -20, & b'_1 = -46,96; \\
 b_2 = -52,96, & b'_2 = -66,26; \\
 b_3 = -76,26, & b'_3 = 11113,8; \\
 b_4 = 11107,8, & b'_4 = 74,7.
 \end{array}$$

Pro příčné zvětšení dostaneme $\beta_B = -2,88$. Zbývá jen dopočítat osové zvětšení dle 35, dostaneme $\alpha = -7,69$. Tím máme druhý příklad vyřešen. Při řešení s použitím knihovny dostaneme shodné výsledky. Přidejme ještě schéma, viz obr. 22, průchodu dvou zadaných paprsků zadanou optickou soustavou.



Obrázek 22 | Zobrazení dvou předmětů skrz optickou soustavou z 4.2.

4.3. 3. příklad

Najděte souřadnice obrazu předmětu A , jestliže je vzdálenost jeho kolmého průmětu do optické osy a vrcholu první lámavé plochy 1,5, po zobrazení optickou soustavou:

$$\begin{array}{lll} r_1 = 0,3486, & & n_1 = 1; \\ r_2 = -0,3664, & d_1 = 0,06, & n'_1 = n_2 = 1,52; \\ r_3 = -0,2958, & d_2 = 0,1, & n'_2 = n_3 = 1; \\ r_4 = 3,5924, & d_3 = 0,02, & n'_3 = n_4 = 1,63; \\ r_5 = -0,3242, & d_4 = 0,1, & n'_4 = n_5 = 1; \\ r_6 = -0,3655, & d_5 = 0,03, & n'_5 = n_6 = 1,63; \\ r_7 = -0,5746, & d_6 = 0,1, & n'_6 = n_7 = 1; \\ r_8 = -0,7465, & d_7 = 0,03, & n'_8 = n_9 = 1,52; \\ & & n'_9 = 1. \end{array}$$

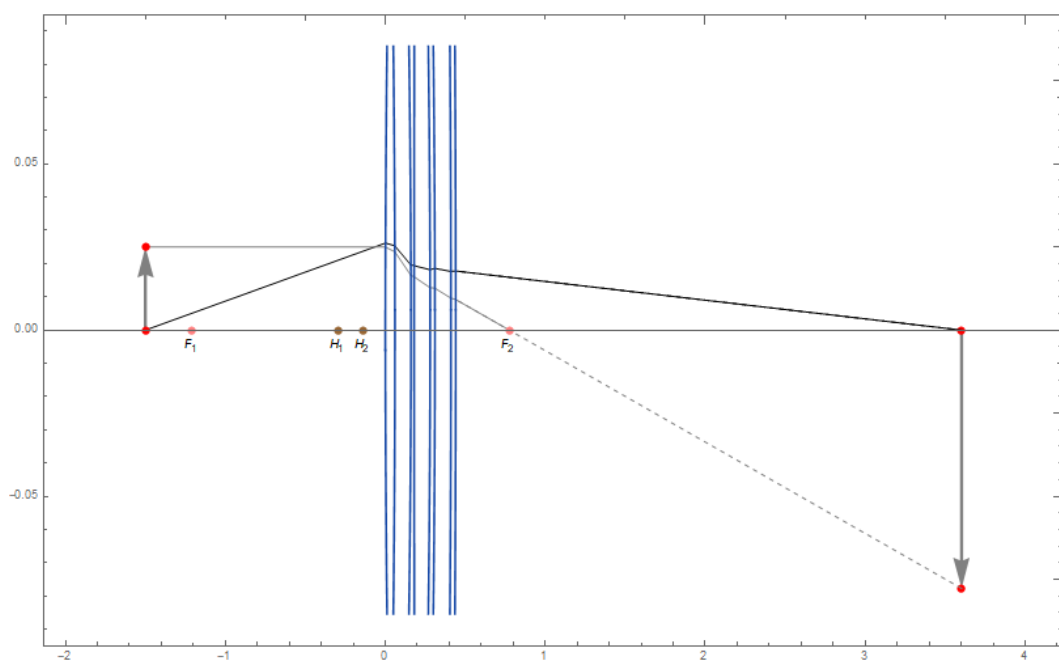
Vzdálenost předmětu od optické osy je 0,025. Předmět se nachází před první lámavou plochou. Počátek soustavy souřadné se nachází ve vrcholu první lámavé plochy a optická osa je shodná s osou x .

K vyřešení problému musíme najít bod, který bude průmětem obrazu do optické osy. A dále také zvětšení zadaného bodu. Pro oba výpočty potřebujeme znát dílčí předmětové a obrazové vzdálenosti, postupujeme jako v předchozích dvou příkladech:

$$\begin{array}{ll} a_1 = -1,5, & a'_1 = 1,84; \\ a_2 = 1,78, & a'_2 = 0,44; \\ a_3 = 0,34, & a'_3 = 2,01; \\ a_4 = 1,99, & a'_4 = 1,56; \\ a_5 = 1,46, & a'_5 = -1,3; \\ a_6 = -1,33, & a'_6 = 2,02; \\ a_7 = 1,92, & a'_7 = -3,97; \\ a_8 = -4, & a'_8 = 3,16. \end{array}$$

Zjistili jsme, že vzdálenost poslední lámavé plochy a kolmého průmětu obrazu do optické osy je a'_8 . Sečtením vzdáleností mezi lámavými plochami a obrazové vzdálenosti a'_8 dostaneme souřadnici obrazu v ose x .

Dle 34 určíme zvětšení $\beta = -3,11$. Souřadnice obrazu A' předmětu A jsou $[3,6;-0,78]$. V prostředí Wolfram Mathematica je řešení příkladu téměř shodné s předchozími. Vytvoříme reprezentaci optické soustavy, užijeme funkci `ThneedInvention[]`. Dále zobrazíme dosažené výsledky, abychom měli představu o chodu paprsku, viz 23.



Obrázek 23 | Zobrazení předmětu skrz optickou soustavou z 4.3.

Závěr

Při popisu build-in výrazů prostředí Mathematica jsme se omezili na stručný popis funkcí a parametrů nezbytných pro vytváření modelů a knihovny, protože každá jedna funkce je velmi podrobně popsána v dokumentaci jazyka Wolfram, tj. [10].

Vytvořili jsme knihovnu funkcí, určenou pro vizualizaci optických zobrazení, ze kterých můžeme vytvořit dynamické i statické modely průchodu paraxiálního paprsku optickou soustavou složenou ze sférických lámavých ploch. U každé funkce je uveden přesný popis vstupních parametrů i výstupů a nástin problémů řešených při konstrukci těchto funkcí. Příkladem modelů vytvořených pomocí knihovny jsou jak rozhraní a dynamický model optické lavice, tak simulace chodu rovnoběžných paprsků optickou soustavou, příklad zobrazení většího množství předmětů i tři klasické příklady řešené na konci práce.

Rozhraní slouží k vizualizaci optických soustav složených až ze čtyř tlustých čoček a zobrazení nejvýše dvou předmětů. Toto omezení je na rozhraní kladeno z důvodu zachování jednoduchosti. Jelikož má být toto rozhraní užíváno studenty, je nezbytné aby sledovali změnu chodu paprsku optickou soustavou namísto zmatenosti ze složitějšího rozhraní. Podkapitola 3.4 tvoří návod k použití rozhraní. Pro vizualizaci složitějších problémů nebo vypsání většího množství informací o soustavě či zobrazovaném předmětu je nutná aplikace knihovny. Příklady těchto aplikací jsou zmíněny ve třetí kapitole a kód je dostupný na přiloženém CD.

Tři řešené příklady uvádíme ve 4. kapitole. Na výsledcích můžeme vidět, že se vypočtené hodnoty neliší od hodnot, které byly spočteny pomocí knihovny v prostředí Mathematica. Jedná se o klasické příklady. Typově obdobné příklady mohou být řešeny v úvodních kurzech optiky na Přírodovědecké fakultě Univerzity Palackého.

Použití naší knihovny a obecně prostředí Wolfram Mathematica na středních školách by mohlo napomoci k integraci rozličných předmětů, respektive k prorůstání informatiky do přírodovědných předmětů a matematiky, díky jednoduchosti základního použití prostředí Mathematica. Pro většinu škol by bylo sehnání licencí na prostředí Mathematica problematické, protože na rozdíl od Pythonu nebo Octave není dostupné zdarma. Avšak na webu <https://wolframcloud.com/> je dostupná omezená online verze. Hlavní nevýhodou je nemožnost nahrání vlastního souboru .nb.

Vystavení dynamického modelu optické lavice v Pevnosti poznání vedle modelu fyzického by mohlo návštěvníkům posloužit k pochopení chování soustavy dvou tlustých čoček. V expozici Světlo a tma se nacházejí i počítače, tzv. kiosky, na kterých by mohl model běžet.

Společnost Wolfram provozuje webovou stránku [11] s dynamickými modely vytvořenými v jazyce Wolfram nazvanou Wolfram Demonstration Project. Dynamické vizualizace vytvořené pomocí naší knihovny by mohly být na tyto stránky nahrány. Bohužel kód vytvořeného rozhraní nesplňuje některé požadavky na umístění. Konkrétně se jedná o nemožnost použití dvou do sebe zanořených funkcí `Manipulate[]`. V době psaní tohoto textu se na stránkách Wolfram Demonstration project nenachází dynamický model mající takové možnosti jako vytvořené rozhraní. V [11] jsou zveřejněny simulace odrazu paprsku na kulovém zrcadle, simulace chodu paprsku jednou či dvěma tlustými čočkami a velké množství vizualizací z oblasti optiky, které se netýkají našeho tématu.

Na bakalářskou práci by bylo možné navázat rozšířením knihovny o zrcadla, asférické lámavé plochy, optické prvky s nekonztantním indexem lomu, ale též o přidání vizualizace vad optických zobrazení a trasování paprsků ve trojrozměrném prostoru. Potom by modely vytvořené pomocí knihovny ještě lépe posloužily k pochopení principů paprskové optiky.

Seznam použitých zdrojů

- [1] WOLFRAM, Stephen. *An elementary introduction to the Wolfram language*. Champaign, IL, USA: Wolfram Media, [2015]. ISBN 1944183000.
- [2] REICHL, Jaroslav. *Optické zobrazení*. Encyklopedie fyziky [online]. 2007 [cit. 2019-07-22]. Dostupné z: <http://fyzika.jreichl.com/main.article/view/472-opticke-zobrazeni>
- [3] VON LAUE, Max. *Dějiny fyziky*. Praha: Orbis, 1959.
- [4] ŠTOLL, Ivan. *Dějiny fyziky*. Praha: Prometheus, 2009. ISBN 978-80-7196-375-2.
- [5] FEYNMAN, Richard Phillips, Robert B. LEIGHTON a Matthew SANDS. *Feynmanovy přednášky z fyziky s řešenými příklady 1*. Havlíčkův Brod: Fragment, 2000. ISBN 80-7200-405-0.
- [6] BAJER, Jiří. *Optika 1*. Olomouc: Univerzita Palackého v Olomouci, 2018. ISBN 978-80-244-4532-8.
- [7] HAVELKA, Bedřich. *Geometrická optika*. Praha: ČSAV, 1955.
- [8] BROOKER, Geoffrey. *Modern classical optics*. New York: Oxford University Press, 2003. ISBN 978-0-19-859964-7.
- [9] *Stephen Wolfram* [online]. Wolfram [cit. 2019-07-31]. Dostupné z: <https://www.stephenwolfram.com/>
- [10] *Wolfram Language & System Documentation Center* [online]. Wolfram, c2019 [cit. 2019-07-31]. Dostupné z: <https://reference.wolfram.com/language/>
- [11] Optics. *Wolfram Demonstration Project* [online]. Wolfram Demonstrations Project & Contributors, c2019 [cit. 2019-08-05]. Dostupné z: <https://demonstrations.wolfram.com/topic.html?topic=Optics&limit=20>

Příloha 1. Kód na CD

K bakalářské práci je přiloženo CD se souborem .nb, který obsahuje kód knihovny i příkladů, a elektronickou verzi textu práce v .pdf.