Scientific
Wizard Ltd.

Numerical Generalization of the Linear Dispersion Compound Prism Spectrometer

ABSTRACT

Prism based spectrometers are utilizing the wavelength dispersive characteristics of the prism. Their main advantage over grating based spectrometers is the lack of high-order phenomena which lead to ghosts and cross-talk. But the main limit of prisms was always the non-linear behavior of angle vs. wavelength.

In a previous paper the Compound Prism Spectrometer was introduced, while a specific optical design was used to demonstrate the concept.

In this paper a numerical generalization is performed, to prove the robustness of the design, and the ability to use virtually any 2 kinds of glasses.

The model described below, includes a bank of Schott glasses, and the equations are built to find and optimize two parameters: dispersion linearity and dispersion span

Keywords: Linear Spectral Dispersion, Compound Prism, Numerical Model

1. Introduction

Research essentials are presented on the following pages.

Introduction, conclusions, and references - will be completed soon.

In general we are looking for the solution of the equation:

$f(\lambda) = a\sqrt{h(\lambda)} + b\sqrt{g(\lambda)}$:

We are looking to find the ratio between ***a*** and ***b*** in such way that $f(\lambda)$ will be linear in given range.
The rules over ***a*** and ***b*** are that ***a*** is positive and in the range of 1 to 45 and ***b*** is always negative in the same range.
The range for λ is given by the physical transparency of the glass.
In this specific solution we use Schott definition for refraction index as function of λ.
The spectral range is set in accordance to visible light range i.e. 0.4 to 1 micron.
Using Schott definition the index of refraction is given by:

$$n_{schott} := \mathrm{sqrt}\left(\frac{K_1 \cdot \lambda^2}{\lambda^2 - L_1} + \frac{K_2 \cdot \lambda^2}{\lambda^2 - L_2} + \frac{K_3 \cdot \lambda^2}{\lambda^2 - L_3} + 1\right) :$$

We prove the concept that the refraction index of combined prisms of different glasses can be described by linear function. We use two glasses, one is SF11 and the other is KDP.
We will start with the coefficients of SF11:

$Kgroup := \{K_1 = 1.73759695\mathrm{E}+00, K_2 = 3.13747346\mathrm{E}\text{-}01, K_3$
$= 1.89878101\mathrm{E}+00, L_1 = 1.31887070\mathrm{E}\text{-}02, L_2 = 6.23068142\mathrm{E}\text{-}02, L_3$
$= 1.55236290\mathrm{E}+02\}$:

We will assign the SF11 coefficients into the general function of the refraction index:

$SF11 := subs\left(Kgroup, n_{schott}\right)$:

For KPD the coefficients are:

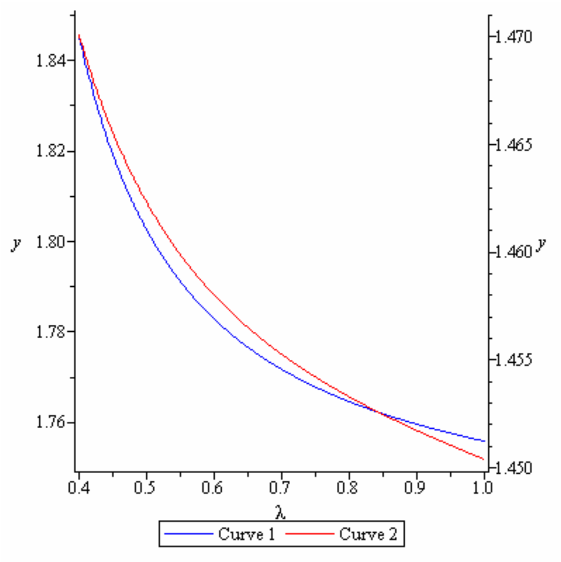
$Kgroup2 := \{K_1 = 6.70710810\mathrm{E}\text{-}01, K_2 = 4.33322857\mathrm{E}\text{-}01, K_3$
$= 8.77379057\mathrm{E}\text{-}01, L_1 = 4.49192312\mathrm{E}\text{-}03, L_2 = 1.32812976\mathrm{E}\text{-}02, L_3$
$= 9.58899878\mathrm{E}+01\}$:

Following the previous step we apply:

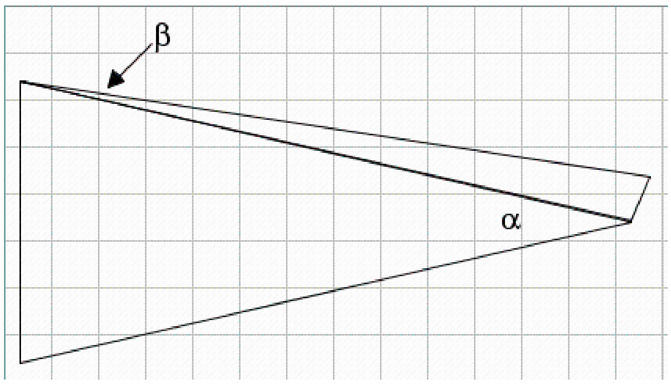
$NGlass := subs\left(Kgroup2, n_{schott}\right)$:

We will plot the refraction index of SF11 and KPD as function of λ.

$G1 := plot(SF11, \lambda = 0.4..1, y = 1.75..1.85, color = blue)$:
$G2 := plot(NGlass, \lambda = 0.4..1, y = 1.45..1.471)$:
$plots[dualaxisplot](G1, G2)$



Two prisms of different glasses have a refraction index that is a linear combination of the refraction index of each glass. The parameters that rule the final results are the head angle of each prism. There for we get $n_{new} := \alpha \cdot n_1 + \beta \cdot n_2$:



Our claim is that there is ration between α and β that will make the combined refraction index linear as function of λ. The for we will define β as function of α and R, the ratio factor $\beta_1 := -\frac{\alpha_1}{R_1}$:

The private case of SF11 and KDP will show that this result can be obtained:

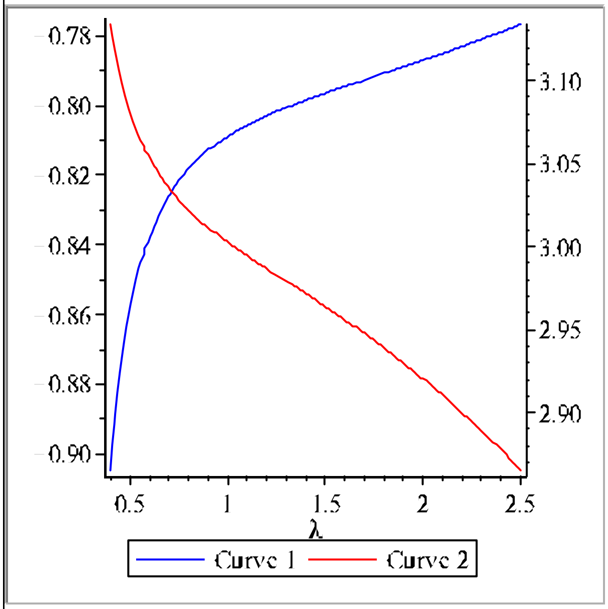
$SF11inP := \beta_1 \cdot (SF11 - 1)$:

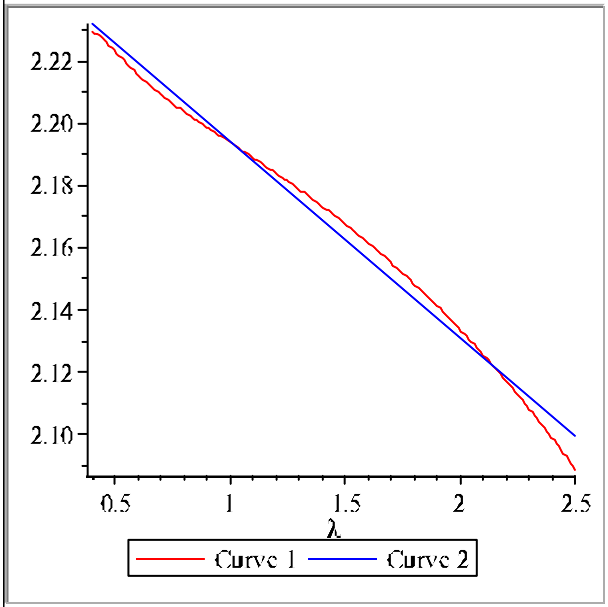
$\mathrm{NGlassinP} := \alpha_1 \cdot (NGlass - 1)$:

$eq1 := SF11inP + \mathrm{NGlassinP}$:

Using the dial buttons one can see the effect of R and α on the refraction index of each part of the optical system and the refraction index of the combined optical system. In blue there is a linear fit to the result.

One can see that the effect of α on the combined graph is changing the "DC" value of the graph, while changes of R give changes of the function behavior in the given range.





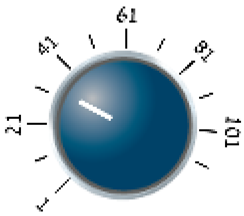

```
HFloat(2.2574247671683776)-HFloat(0.06323612159166415)*l
```

Read Schott data:

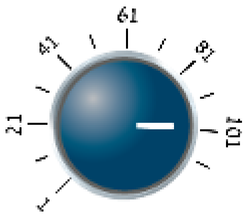
After proving the private case of SF11 and KPD we will go to any set of two glasses. A list of glass made by Schott was obtained. The list is loaded and using the buttons one can load different pairs of glasses, then change α and R and see the effect. The result is that any pair in the given list has on R that makes the combined graph linear. In order to find the result is linear we find the slop of the linear fit and the derivative of the combined refraction index at two points (0.5 and 0.8). When the three values are identical (or close to equal) the response of the refraction index to λ is linear.

```
-7.550648866*abs(1.31004128*lambda^2/(lambda^2-0.79659645e-2)+.142038259*lambda^2/(lambda^2-0.330672072e-1)+.964929351*
/(lambda^2-109.19732)+1)^(1/2)+3.666134666+3.8845142*abs(1.84910553*lambda^2/(lambda^2-0.999234757e-2)+.329828674*lambda
bda^2-0.387437988e-1)+1.30400901*lambda^2/(lambda^2-95.8967681)+1)^(1/2)
```

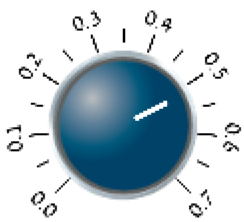
```
0.204649989e-1    0.180720391e-1    HFloat(0.018691604945055982)
```



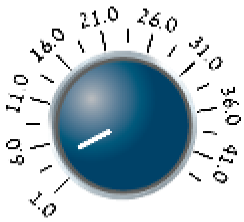
N-BALF4



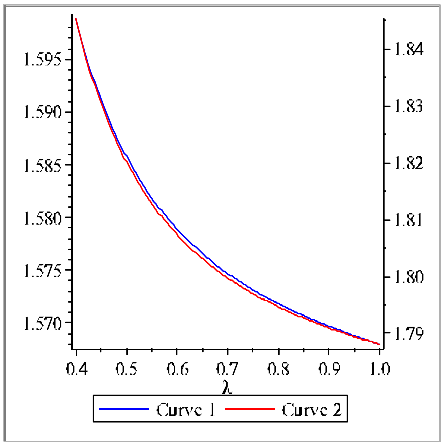
P-LASF50

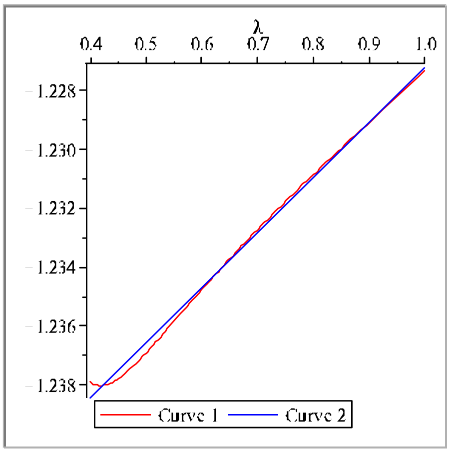


.514460978



3.8845142





As final step we developed a numerical algorithm that can find R and the dispersion, for any given pair of glasses.

Annex A – code in brief

```
for G1 from 1 to N do
KG1 := {K_1 = C(G1, 2), K_2 = C(G1, 3), K_3 = C(G1, 4), L_1 = C(G1,
    5), L_2 = C(G1, 6), L_3 = C(G1, 7)} :
KGG1 := abs(subs(KG1, n_schott)) :
 for G2 from 120 − N to 119 do
Min1 := 100 :
Min2 := 100 :
Min1Last := 101 :
Min2Last := 101 :
KG2 := {K_1 = C(G2, 2), K_2 = C(G2, 3), K_3 = C(G2, 4), L_1 = C(G2,
    5), L_2 = C(G2, 6), L_3 = C(G2, 7)} :
KGG2 := abs(subs(KG2, n_schott)) :
for R from 0.01 by 0.01 to 10 while (Min1 < Min1Last and Min2
    < Min2Last) do
Min1Last := Min1 :
Min2Last := Min2 :
β := -α/R :
G1P := β*(KGG1 − 1) :
G2P := α*(KGG2 − 1) :
eqGT := G1P + G2P :
 F2 := convert([ seq(eqGT, λ = StartP .. EndP, 0.01)], Vector) :
 X1 := convert([ seq(i, i = StartP .. EndP, 0.01)], Vector) :
 Dif1 := eval(diff(eqGT, λ), λ = TestP1) :
 Dif2 := eval(diff(eqGT, λ), λ = TestP2) :
Z := LinearFit([1, λ], X1, F2, λ) :
DifZ := diff(Z, λ) :
D1 := abs(DifZ − Dif1) :
D2 := abs(DifZ − Dif2) :
Min1 := min(Min1, D1);
Min2 := min(Min2, D2);
Out1[i, j] := R;
Out2[i, j] := DifZ;
end do:
i := i + 1;
end do:
 i := 1;
j := j + 1;
end do:
 max(Out1);
 max(Out2);
```