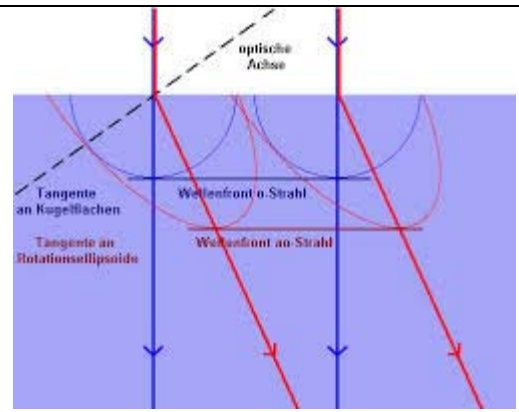


### Aufgabenblatt 3

#### I) Gelerntes wiedergeben

- Erklären Sie anhand des nebenstehenden Bildes, warum man „doppelt“ sieht, wenn man durch Kalkspat blickt. (vgl. Vorlesungsexperiment)
- Wie muss man einen Kalkspat zur Blickrichtung orientieren, damit man NICHT doppelt sieht? Es wird einfach, wenn Sie sich gestatten, den Kristall dafür speziell zuzuschneiden. Wie?
- Skizzieren Sie die Reflektivität einer Vakuum-Glas Grenzfläche als Funktion des Einfallswinkels für parallel und senkrecht polarisiertes Licht.  $n_{\text{Glas}} = 1.5$ .



- Skizzieren Sie die Reflektivität einer Glas-Vakuum Grenzfläche als Funktion des Einfallswinkels für parallel und senkrecht polarisiertes Licht.  $n_{\text{Glas}} = 1.5$ .

#### II) Einfache Aufgaben

##### II.4) Absorption

Ein Lichtstrahl der Wellenlänge  $\lambda = 400 \text{ nm}$  und Intensität  $I_0$  trifft aus dem Vakuum senkrecht auf eine Grenzfläche mit einem Material, dessen komplexer Brechungsindex  $\tilde{n} = n + i\kappa = 1.52 + i 1.84$  ist.

- Wie groß ist die Wellenlänge  $\lambda_m$  im Material?
- Nach welcher Propagationslänge  $L$  ist die Intensität auf die Hälfte abgefallen? Wie vielen Wellenlängen  $\lambda_m$  entspricht das?

##### II.5) Brewsterwinkel

Leiten Sie aus den Fresnelschen Formeln eine Gleichung ab, die beschreibt, unter welchem Winkel kein parallel zur Einfallsebene polarisiertes Licht von einer Grenzfläche reflektiert wird, die zwei beliebige homogene Medien mit reellem Brechungsindex  $n_1$  und  $n_2$  trennt. Gibt es diesen Brewsterwinkel für  $n_1 < n_2$  und  $n_1 > n_2$ ?

##### \*II.6.) Dielektrische Funktionen

- Plotten Sie den Real- und Imaginärteil einer dielektrischen Funktion mit zwei Resonanzen. Wählen Sie die Oszillatorstärken, Frequenzen und Dämpfungskonstanten so, dass man auf einer logarithmischen Frequenzachse ein Bild sieht, das dem der Zusammenfassung ähnelt. Sie dürfen jedes vernünftige Mathematik-Programm nutzen. Wouter Koopman (Seite 2) hat eine Kurzanleitung zum Installieren und Verwenden von Python verfasst... Dauert nur wenige Minuten sich einzuarbeiten.
- Falls Sie das Python Programm verwenden, schreiben Sie es zur Übung um, indem Sie die Notation „c.real“ verwenden, um den Realteil einer komplexen Zahl zu erhalten. Achtung: In Python wird die komplexe Zahl  $i$  als  $j$  dargestellt. Versuchen Sie mal unterschiedliche Möglichkeiten... Rumprobieren...
- Wie groß ist die Differenz zwischen dem Wert des Realteils deutlich ober- und unterhalb der Resonanz? Probieren und Formel aufstellen! Hängt diese Differenz von der Dämpfung ab?

#### III) Vertiefende Aufgaben

##### \*III.5) Fresnelsche Formeln

- Skizzieren Sie ein Strahlenbündel das von einem Medium mit Brechungsindex  $n_1$  in ein Medium mit höherem Brechungsindex  $n_2$  eintritt. Dabei ändert sich der Bündelquerschnitt von  $A_1$  auf  $A_2$ .
- Der Energiestrom des einfallenden Strahls ist  $\dot{W}_e = c_1 \rho_e A_1$ , der des reflektierten Strahls ist  $\dot{W}_r = c_1 \rho_r A_1$ , und der des transmittierten Strahls  $\dot{W}_t = c_2 \rho_t A_2$ . Stellen Sie eine Gleichung auf, die besagt, dass der einfallende Energiestrom gleich der Summe des reflektierten und transmittierten Energiestroms ist. Drücken Sie dabei die Flächen der Bündel durch ein gemeinsames  $A$  und den Einfallswinkel  $\alpha_1$  und Brechungswinkel  $\alpha_2$  aus, verwenden Sie  $c_{1,2} = n_{1,2} c_0$  für die Lichtgeschwindigkeit und benutzen Sie für die Energiedichte  $\rho = \frac{1}{2} \epsilon_0 \epsilon E^2$  mit geeignetem Subskript.

- Wenn Sie jetzt noch  $\epsilon_{1,2} = n_{1,2}^2$  verwenden erhalten Sie eine Gleichung der Form:

$$\underline{\quad} * (E_e^2 - E_r^2) * \cos \alpha_1 = \underline{\quad} * E_t^2 * \cos \alpha_2$$

- Für senkrechte Polarisation der E-Felder besagt die Stetigkeit der Tangentialkomponente der Feldstärke:  $E_e + E_r = E_d$ . Warum? Leiten Sie daraus die Fresnelgleichung für  $\frac{E_e^2}{E_t^2}$  her.

## Datenverarbeitung

Nebst einer Reihe von kommerziellen Programmen (Origin, etc.) und Programmiersprachen (MatLab, MatCAD, Mathematica) hat sich in den letzten Jahren die freie Programmiersprache Python zum Quasi-Standard für die wissenschaftlichen Datenverarbeitung entwickelt. Die Grundlage für die Verarbeitung von Daten bilden die Python-Module **NumPy** (*Numeric Python*) und **SciPy** (*Scientific Python*). Ersteres erlaubt die Verarbeitung und Manipulation großer, multidimensionaler (Daten-)Arrays, während letzteres viele Funktionen zur Datenverarbeitung aus Bereichen wie Optimierung/Fitting, Bild- und Signalverarbeitung, (numerische) Integration und Differentiation, lineare Algebra, FFT u.v.m. bietet. Zur graphischen Darstellung von Daten kann z.B. Matplotlib verwendet werden, das umfangreiche Möglichkeiten zur Darstellung von 2D Plots und Bilder bietet.

Die einfachste Art und Weise Python zur Datenverarbeitung zu auf einen lokalen Rechner zu installieren, ist es das Installationspaket Anaconda zu verwenden, welches Continuum Analytics als Open-Source Software zur Verfügung stellt. Das Paket kann unter:

<https://www.continuum.io/downloads>

heruntergeladen werden. Eine kurze Anleitung (Englisch) zur Installation gibt es unter:

<https://docs.continuum.io/anaconda/install>

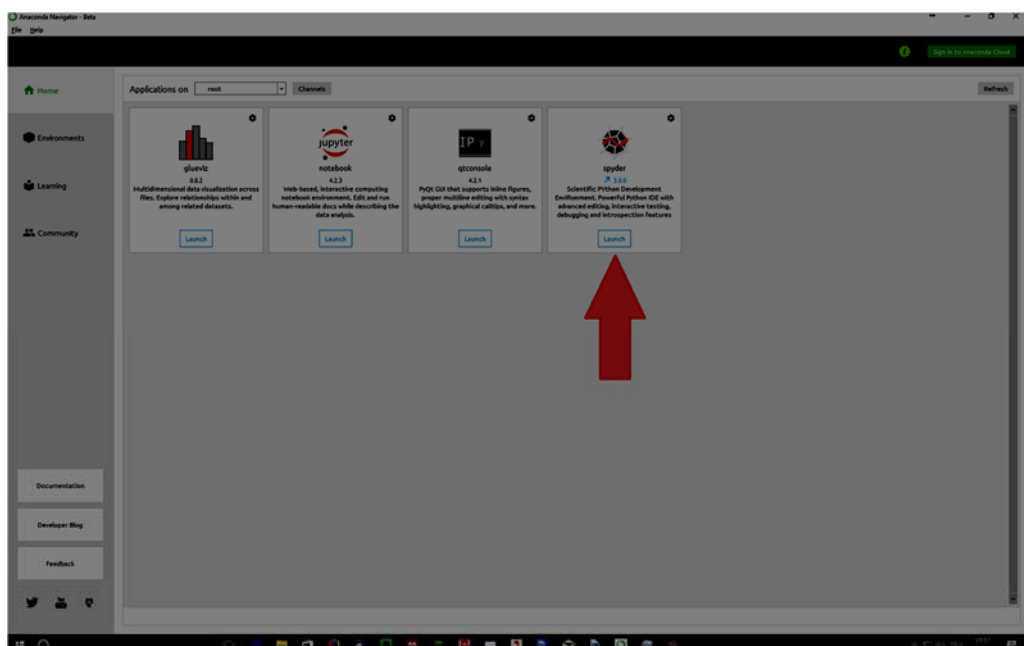
Für Interessierte gibt es Tutorials zu NumPy und Matplotlib unter:

NumPy Documentation: <https://docs.scipy.org/doc/numpy/user/index.html>

Matplotlib Beginners Guide: <http://matplotlib.org/users/beginner.html>

Zur Verwendung von Python reicht in Prinzip ein einfacher Texteditor, sowie die Kommandozeile. Wesentlich bequemer ist aber die Verwendung eines IDE (Integrated Development Environment). Für die Datenverarbeitung mit Python bietet sich hier **Spyder** an, das in der Anaconda installation bereits enthalten ist und durch den **Anaconda Navigator** gestartet werden kann.

Die wichtigsten Elementen von Spyder sind das Konsolen (IPython Console) und das Editor Fenster.



Da Python eine *Interpreter-basierte Programmiersprache* ist (ähnlich wie z.B. MatLab), können in der Konsole einzelne Python-Befehle eingegeben und ausgeführt werden. Im Editor können wiederum ganze Programme geschrieben (sog. Skripte und Module im Python-Jargon). Das gesamte Modul kann in die Konsole geladen werden, entweder durch betätigen des „Run File“ Symbols in der Symbolleiste, oder durch Drücken der F5 Taste. Einzelne Funktionen des Moduls können dann in der Konsole verwendet werden.

Zum Laden der Simulation zur Dispersionsrelation, muss zunächst das Python-Skript in den Editor geladen werden (Datei > öffnen > „ExIII\_Lorentz.py“). Vorher müssen Sie diese Datei von der teaching-homepage der udkm laden. Leider liegt sie dort als ExIII\_Lorentz.pdf. Sie müssen bitte die Endung .pdf durch .py ersetzen. Das Programm wird dann durch Drücken der F5 Taste in die Konsole geladen. Die Dielektrische Funktion kann dann durch Eingabe des Funktionsnamen

```
permittivity()
```

in die Konsole gestartet werden. Es erscheinen zwei Fenster, die den Real- und Imaginärteil der Permittivität grafisch darstellen. Optional können der Funktion die Eigenfrequenzen und Dämpfungen der beiden Oszillatoren mit Hilfe von Signalwörtern übergeben werden:

```
permittivity(osc1=300, gamma1=100, osc2 = 1200, gamma2=100)
```