Juan Pedro Martinez jmartinez@fing.edu.uy
Website: https://juanmartinezalvarez.github.io/

MATLAB Transfer Matrix Method Code Guide for the Calculation of Transmission and Reflection in Isotropic Stratified Media

If you include these codes in your work, please credit as: Martinez, J. P. Light propagation in multilayered nanostructures. (2024) doi:10.13140/RG.2.2.30332.96640.

1. The codes in the main folder are required to run the programs. The scripts I'm going to comment on use the others as input. It is important that they are in the Matlab path, the easiest thing is to have them all in the same folder
2. **The convention for the refractive index is $\hat{n} = n - ik$. This is equivalent to the $e^{i(wt-kx)}$ convention for the plane wave. (positive k gives decreasing exponential)**

Fresnel Interface

This code takes as input an angle of incidence and a vector of refractive indices (it can be complex) and gives you the Fresnel coefficients at each interface, along with the propagation angles within each medium.

```
[rs,ts,rp,tp,phi]=fresnel_interface(n,phi0)
```

- **n**: row vector of dimensions 1x(N+2) (N is the number of layers) with the refractive indices in the order in which light finds them. You must include the incoming and outgoing media (the "environment"). For example, if I want to model a system composed of two layers: Silicon on glass, silicon on top, glass resting on water (outgoing media), and air in the entrance (incoming media), the vector is n= $[1, n_{Si}, n_{glass}, n_{water}]$
- **phi0**: angle of incidence in **DEGREES**
- **rs:** vector with the Fresnel reflection coefficient for the s-polarized wave (perpendicular to the plane of incidence, also called transverse electric, TE) each element corresponds to each interface, in order. In the last case:

  $$\left[rs_{air-Si},\ rs_{Si-glass}, rs_{glass-water}\right]$$

  This vector has dimensions N+1

- ts,rp,tp: same as before put for transmittance (t) and the p polarized wave (parallel to the plane of incidence, transverse magnetic, TM)
- **Phi:** vector with the propagation angle in each media (in order), calculated from Snell's law. The first element is phi0, the last one is the angle in transmittance from the multilayered structure.vector con el ángulo de propagación en cada interfaz, calculado con la ley de Snell. Vector of dimensión N+2, The vector is in **RADIANS**.

Juan Pedro Martinez jmartinez@fing.edu.uy
Website: https://juanmartinezalvarez.github.io/

Transfer matrix method

This function gives the reflectance (R) and transmittance (T).

```
[Values]=TMM_fresnel_inc_file_vec(n,e,phi0,lambda,coh)%coh=1 if
layer is incoherent
```

- **`n:`** is a **CELL ARRAY**, a data structure in MATLAB that uses {} as brackets, you can include text (string) and numbers to that structure. It is a vector of 1x(N+2) as before, composed of the refractive indices. The only difference is that you can also include a file with the tabulated values, to make it spectral (see comment on lambda). The file .txt to be included, which has the spectral dependence of the refractive index of a material in a layer, must be in the Matlab workspace, and it must have 3 columns: λ, n and k. In that order. No headers should be included. In the rows, the wavelength varies. If the material does not absorb, i.e. k=0, you must put in the third column all 0s. The wavelengths to calculate for (lambda, see below) must be in the file, the code does not interpolate.
- **`e:`** vector with the thickness of each layer, dimensión 1xN, as the incoming and outgoing media are semi-infinite, meaning they have no thickness. The thicknesses must go in **nanometers (nm)**.
- **`phi0:`** Angle of incidence in **DEGREES**. It can be a single number or a vector if the calculations are to be performed as a function of angle of incidence, resulting in Rs(ϕ).
- **`Lambda:`** A vector with wavelengths. It can be a single number (for one wavelength) or a vector if you want to do it spectrally and obtain Rs(λ). If you do not provide a file for any of the elements of n, the spectral dependence only plays a role in the propagation within a medium (modifies absorption). If you provide a file, the program goes to the file and looks up the wavelength, defines n-ik for that wavelength, and calculates. The wavelengths in the lambda vector must **EXACTLY** match those in the file. The program does not perform any interpolation
  - **Note:** You can pass phi0 or lambda as a vector, but not both at the same time. If one is a vector, the other must be a scalar. Both can be scalars.
- **`Coh:`** A vector that indexes the layers and indicates whether they are coherent or not. A 0 means that the propagation is coherent, and a 1 means it is incoherent. There is one value per layer, so it has a dimension of 1xN.
- **`Values:`** It is a MATRIX with 4 columns, which are, in this order: Rs, Ts, Rp, Tp. The rows correspond to each value of phi0 or lambda, depending on which one was entered as a vector. If both are scalars, this Values vector is 1x4.

<u>Example:</u> Silicon on glass, with silicon on top, glass supported on water, and air as the entry medium. For silicon, I want to have the spectral dependence of the refractive index, so I use a file called 'Si_nyk.txt'. The thickness of silicon is 200nm, and the thickness of glass is 10000nm. Glass is incoherent. Incident wavelength is 600nm.

Case 1: Calculate with angular dependence

[Values]=TMM_fresnel_inc_file_vec($[1,'Si\_nyk.txt', n_{glass}, n_{water}]$,[200,10000],[0,20,45],600,[0,1])

Juan Pedro Martinez jmartinez@fing.edu.uy
Website: https://juanmartinezalvarez.github.io/

$$Values = \begin{bmatrix} Rs(0) & Ts(0) & Rp(0) & Tp(0) \\ Rs(20) & Ts(20) & Rp(20) & Tp(20) \\ Rs(45) & Ts(45) & Rp(45) & Tp(45) \end{bmatrix}$$

Case 2: spectral dependence

[Values]=TMM_fresnel_inc_file_vec($[1,'Si\_nyk.txt', n_{glass}, n_{water}]$,[200,10000],0,[450,600,800],[0,1])

$$Values = \begin{bmatrix} Rs(450) & Ts(450) & Rp(450) & Tp(450) \\ Rs(600) & Ts(600) & Rp(600) & Tp(600) \\ Rs(800) & Ts(800) & Rp(800) & Tp(800) \end{bmatrix}$$

Modelo matriz de transferencia CAMPOS

This function is the same as before (`TMM_fresnel_inc_file_ vec`), except the outputs are directly the fields, not the intensities. This is useful for example to calculate the Stokes parameters or the ellipsometric parameters. To work properly ALL layers must be coherent, because incoherent layers lose information on fields. Only Works for perfectly coherent media. The output now are each reflection and transmission coefficients independently. They are column vectors, each row corresponds to the wavelength or lambda, if they were vectors.

```
[rs,ts,rp,tp]=TMM_fresnel_inc_file_fields_vec(n,e,phi0,lambda,coh)% n is cell array coh=1 if layer is incoherent
```

Modelo para el cálculo de medios efectivos

This code calculates a mixture of media using the Bruggeman mixing rule. The calculation method, which gives a single root, follows the work of Rouseel (see reference at the end). It mixes two media.

```
[ema]=Bruggeman_Rous_vec(eps1,eps2,c,mode)%c is volumen fraction of material 2%-----
```

- **`eps1:`** Dielectric constant (Note: this is not the refractive index as before; you need to pass the dielectric function epsilon). It can be a vector if you want to perform a mixture for the spectral dependence of the media.
- **`eps2:`** The same as before. If you pass them as vectors, they must both have the same dimension. When one of the media is air, for example, you need to pass a vector of 1s
- **`c:`** is the volumen fraction of component 2 (the second component, the one with eps2). It is a value between 0 and 1.
- **`Mode:`** It is a string that defines the mixing method. There are two options, and they are case-sensitive, etc.
  - Homogeneous mixture of spheres → `'HMS'`

  - Spherical inclusions in a Matrix → 'SIM'

- **ema:** It is a vector that gives the dielectric constant of the mixture. It has the dimensions of the vectors eps1 and eps2. If each of them is a scalar, ema is a scalar. The order of the elements in ema is the same as those of eps1 and eps2

Example: I mix air and silicon, but I want to do it spectrally. The air fraction is 40%, and I mix it as spherical inclusions in the matrix.

[ema]=Bruggeman_Rous_vec([$\in_{Si}$ (450), $\in_{Si}$ (600), $\in_{Si}$ (800)], $[1,1,1]$,**0.4**, 'SIM')

ema=[$\in_{EMA}$ (450), $\in_{EMA}$ (600), $\in_{EMA}$ (800)]

Juan Pedro Martinez jmartinez@fing.edu.uy
Website: https://juanmartinezalvarez.github.io/

References

On the transfer matrix method

Azzam, R. M. A. & Bashara, N. M. (1977). Ellipsometry and polarized light. Amsterdam ; New York : New York : North-Holland Pub. Co. ; sole distributors for the U.S.A. and Canada, Elsevier North-Holland

Página: 332-340

On the transfer matrix method and adding incoherent propagation

E. Centurioni, “Generalized matrix method for calculation of internal light energy flux in mixed coherent and incoherent multilayers,” *Appl. Opt.*, vol. 44, no. 35, p. 7532, Dec. 2005, doi: 10.1364/AO.44.007532.

C. C. Katsidis and D. I. Siapkas, “General transfer-matrix method for optical multilayer systems with coherent, partially coherent, and incoherent interference,” *Appl. Opt.*, vol. 41, no. 19, p. 3978, Jul. 2002, doi: 10.1364/AO.41.003978.

On the method for effectime médium calculations (EMA)

Ph. J. Rouseel, J. Vanhellemont, and H. E. Maes, “Numerical aspects of the implementation of effective-medium approximation models in spectroscopic ellipsometry regression software,” *Thin Solid Films*, vol. 234, no. 1–2, pp. 423–427, Oct. 1993, doi: 10.1016/0040-6090(93)90299-5.