

Exercici 2 d'avaluació Continuada: Interferències d'ones múltiples en làmines plano-paraleles primes.

En aquest exercici s'estudia la convergència de la sèrie geomètrica en les interferències d'ones múltiples en làmines plano-paraleles primes i la seva dependència en el coeficient de reflexió.

Una ona plana de $\lambda = 550 \text{ nm}$ de longitud d'ona incideix sobre una làmina plano-paralela d'un vidre amb un índex de refracció $n = 1.7$ i $7.8 \mu\text{m}$ de gruix. L'ona està polaritzada linealment perpendicular al pla d'incidència.

1. Feu una gràfica del coeficient de reflexió per la component perpendicular al pla d'incidència en funció de l'angle d'incidència en aquest vidre.
2. Feu la gràfica de la intensitat transmesa (un cop travessa tota la làmina) en funció de l'angle d'incidència, considerant el coeficient de reflexió en funció de l'angle (apartat anterior), per la component perpendicular al pla d'incidència. Superposeu en la mateixa gràfica la intensitat transmesa que obtindríem si no tinguéssim en compte els efectes d'interferència. Comenteu les tendències.
3. Ara considerem només un raig que incideix amb 63° respecte de la normal. Calculeu la intensitat transmesa I_T (un cop travessa tota la làmina). L'expressió deduida a classe és fruit d'una sèrie geomètrica. Torneu a calcular la intensitat transmesa tenint en compte únicament les primeres reflexions en l'interior de la làmina. És a dir, sumeu numèricament els $N = 1, 2, 3, \dots, 10$ primers termes de la sèrie geomètrica. Feu una gràfica de I_T en funció del nombre N de termes considerats. Com és la convergència? Per què?
4. Repetiu l'apartat 3. per un raig que incideixi amb 67° respecte de la normal. Convergeix de la mateixa manera que en l'apartat anterior? Per què?
5. Finalment, tingueu en compte únicament les primeres reflexions en l'interior de la làmina. És a dir, com en el punt 1., sumeu numèricament els $N = 1, 2, 3, \dots$ primers termes de la sèrie geomètrica per cada angle d'incidència i, per tant, per cada coeficient de reflexió. El grau de convergència és sempre el mateix? De què depèn?

La nota de l'exercici es calcularà de la següent manera (**tot sobre 1**):

$$Nota = EiP \cdot [0.3 Des. + 0.4 Res. + 0.3 Disc.]$$

	Insuficient (0.0)	Millorable (0.3)	Correcte (0.6)	Molt bé (0.8)	Excel·lent (1.0)
Estructura i presentació (EiP)	El text és confús o incomprensible. L'estructura del document no permet entendre el que es presenta. Fet a mà de qualsevol manera.	El text, tot i ser manuscrit, s'entén amb certes dificultats. Hi ha diversos continguts que no són adequats. L'estructura del document dificulta l'enteniment.	En algunes ocasions el text no és prou clar. En cas que sigui manuscrit, s'entén més o menys. L'estructura del document és adequada.	Es llegeix prou bé, tot i que hi ha certes mancances. L'estructura del document és raonable.	Les explicacions són molt clares, utilitza adequadament la terminologia científica i els apartats estan ben lligats. La presentació del document ajuda a entendre-ho tot plegat.
Desenvolupament (Des.)	No es presenta el desenvolupament teòric i/o traçat de raigs, o conté greus errors.	No té en compte alguna de les relacions més importants entre les diferents magnituds o hi ha errades en les equacions i/o traçats.	El desenvolupament teòric es fa sobre el contingut adequat, tot i que hi ha errors o mancances.	Té en compte la major part de les relacions entre les diferents magnituds i/o els traçats de raigs no contenen errors greus.	Té en compte totes les relacions entre les diferents magnituds i en deriva correctament les equacions i/o els traçats de raigs mostren clarament les relacions entre les magnituds en estudi.
Resultats (Res.)	Els resultats i/o les figures no il·lustren l'objectiu de l'exercici. Les magnituds triades com a més rellevants per representar no són adequades.	Els resultats i/o les figures il·lustren parcialment les tendències més rellevants, però la tria de magnituds per a representar no és del tot adequada.	Tot i fixar unes magnituds i fer variar d'altres de manera adequada, les figures no il·lustren clarament les relacions de les magnituds i els conceptes clau de l'exercici.	Les magnituds representades descriuen prou bé el fenomen estudiat. Tot i així, no són suficients, manca alguna figura per sostenir les conclusions.	Tria adequadament les magnituds a representar i les figures que se'n desprenen són molt descriptives, permeten extreure conclusions.
Discussió (Disc.)	Les conclusions que es descriuen són incorrectes o no s'ajusten als resultats o al model teòric.	Tot i trobar alguna conclusió correcta, no connecta clarament els resultats amb el model teòric.	Connecta correctament alguns resultats calculats amb el model teòric corresponent, però manquen algunes conclusions rellevants.	Connecta correctament els resultats calculats amb el model teòric corresponent.	Connecta de forma clara i justificada els resultats calculats amb el model teòric, tot raonant les tendències trobades.

Nota: Podeu utilitzar qualsevol software/llenguatge per obtenir les gràfiques. En cas que utilitzeu un quadern de Jupyter, podeu entregar-lo directament sense passar-lo a PDF (feu la discussió en cel·les de text). En qualsevol dels altres casos, entregueu només un PDF autocontingut (discussió i gràfics).

```
In [1]: import numpy as np # Llibreria de càlcul numèric i matricial
import matplotlib.pyplot as plt # Llibreria per fer representacions gràfiques
```

Possible solució: NO PUBLICAR fins data de termini d'entrega

```
In [9]: # Dades:
A = 1
n = 1.7
d = 7.8e-6
wavelen = 550e-9

# Explorem els diferents angles d'incidència
theta_1 = np.linspace(0+0.001, np.pi/2-0.001, 1000) # evitem el 0 i el 90°
theta_2 = np.arcsin(1/n * np.sin(theta_1)) # tot va amb l'angle en l'interior del medi

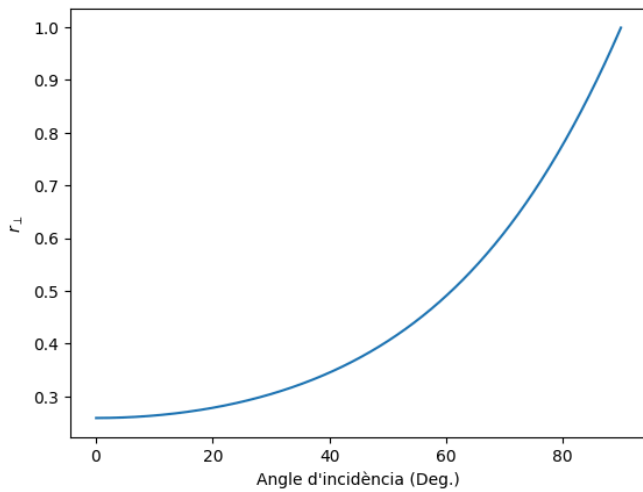
theta_B = np.arctan(n) * 180 / np.pi # Angle de Brewster en graus
```

```
In [52]: # coeficients ortogonals i paral·lel al pla d'incidència
rs = np.sin(theta_1-theta_2) / np.sin(theta_1+theta_2) # coeficient de reflexió
rp = np.tan(theta_1-theta_2) / np.tan(theta_1+theta_2) # coeficient de reflexió

# Factors de reflexió
Rs = rs**2
Rp = rp**2

plt.plot(np.degrees(theta_1), rs)
plt.xlabel("Angle d'incidència (Deg.)")
plt.ylabel("$r_{\perp}$")
```

Out[52]: Text(0, 0.5, '\$r_{\perp}\$')



2.

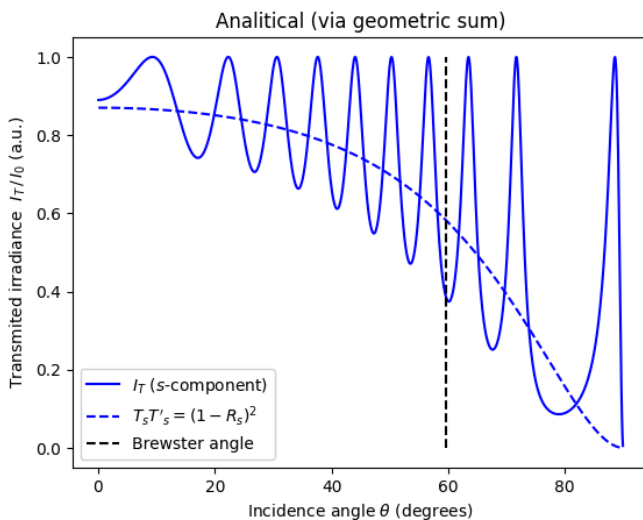
```
In [57]: # Analític

Delta = 2*n*d*np.cos(theta_2)
delta = 2 * np.pi / wavelen * Delta

ITs_geo = A**2 / ( 1 + 4*Rs/(1-Rs)**2 * np.sin(delta/2)**2 )
ITp_geo = A**2 / ( 1 + 4*Rp/(1-Rp)**2 * np.sin(delta/2)**2 )

plt.title("Analitical (via geometric sum)")
plt.plot(theta_1*180/np.pi, ITs_geo, '-b', label="$I_T$ ($s$-component)")
#plt.plot(theta_1*180/np.pi, ITp_geo, '-r', label="$I_T$ ($p$-component)")
plt.plot(theta_1*180/np.pi, (1-Rs)**2, '--b', label="$T_sT'_s = (1-R_s)^2$")
#plt.plot(theta_1*180/np.pi, (1-Rp)**2, '--r', label="$T_pT'_p = (1-R_p)^2$")
plt.plot([theta_B, theta_B], [0, 1], 'k--', label="Brewster angle")
plt.legend()
plt.xlabel(r"Incidence angle $\theta$ (degrees)")
plt.ylabel(r"Transmitted irradiance $I_{T\perp}/I_0$ (a.u.)")
```

Out[57]: Text(0, 0.5, 'Transmitted irradiance \$I_{T\perp}/I_0\$ (a.u.)')



Possible comentari: Podem veure que obtenim pics de transmissió on la transmissió és total. Els mínims no són zero sinó que segueixen la tendència de la corba pel cas de no considerar les reflexions múltiples, i.e. $TT' = (tt')^2 = (1 - R)^2 = (1 - r^2)^2$.

3. i 4.

```
In [56]: def calcula_intensitats(theta1_deg):

    Niter = 10
    theta1_a = np.radians(theta1_deg) # graus
    theta2_a = np.arcsin(1/n * np.sin(theta1_a)) # radians
```

```
Rs_a = ( np.sin(theta1_a-theta2_a) / np.sin(theta1_a+theta2_a) )**2

Delta_a = 2*n*d*np.cos(theta2_a)
delta_a = 2 * np.pi / wavelen * Delta_a

ITp_geo_a = A**2 / ( 1 + 4*Rs_a/(1-Rs_a)**2 * np.sin(delta_a/2)**2 )

field = 0 # inicialitzem el camp electric
ITs_sum = [] # On guardarem les intensitats per cada iteració
for it in range(Niter+1): # +1 per incloure l'últim: range(N)=[0, 1, 2...N-1] ; coses de Python...

    field += A * (1-Rs_a) * Rs_a**it * np.exp(1j*it*delta_a)

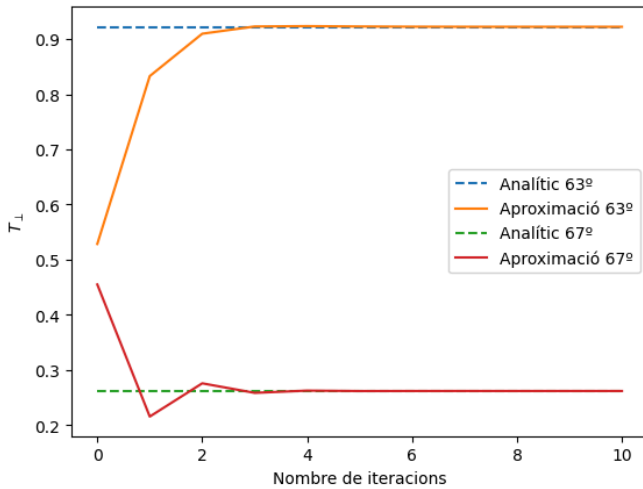
    ITs_sum.append(np.abs(field)**2)

return ITp_geo_a, ITs_sum

ITp_geo_a1, ITs_sum_a1 = calcula_intensitats(63)
ITp_geo_a2, ITs_sum_a2 = calcula_intensitats(67)

plt.plot([0, Niter], [ITp_geo_a1, ITp_geo_a1], '--', label='Analític 63º')
plt.plot(ITs_sum_a1, label='Aproximació 63º')
plt.plot([0, Niter], [ITp_geo_a2, ITp_geo_a2], '--', label='Analític 67º')
plt.plot(ITs_sum_a2, label='Aproximació 67º')
plt.xlabel('Nombre de iteracions')
plt.ylabel("$T_{\perp}$")
plt.legend()
```

Out[56]: <matplotlib.legend.Legend at 0x233c6558e20>



Possible comentari: Per 63° estem prop d'un màxim de transmissió. En un màxim la diferència de camí òptic entre les diferents reflexions és un múltiple de la longitud d'ona. Això fa que totes les contribucions al camp elèctric total van en fase i, per tant, la convergència és asimptòtica (els fasors estan alineats). En canvi, per a 67° estem prop d'un mínim, on la diferència de camí òptic entre dos termes consecutius és un semi-múltiple de la longitud d'ona. Això fa que el signe dels termes s'alterni i la convergència és oscil·lant esmorteïda (els fasors es van alternant i disminuint de magnitud).

5.

```
In [31]: # Aquesta funció sumarà un nombre determinat de termes i els plotejarà, per una R donada
def plot_iterations(R, iterations):

    field = 0 # inicialitzem el camp electric

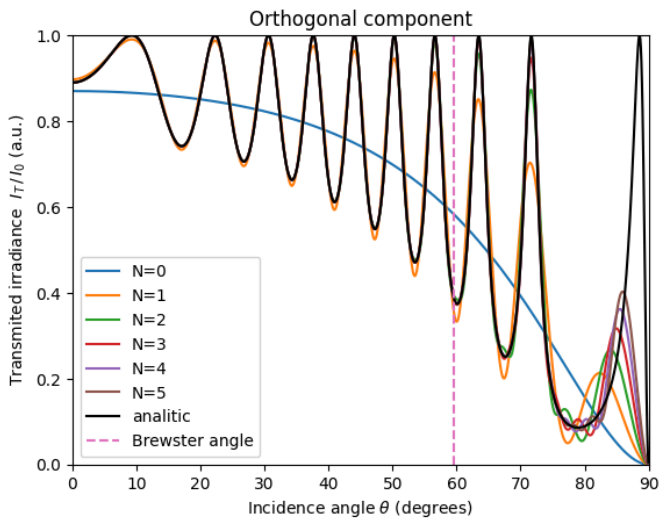
    Ei = [] # On guardarem els camps sumats en cada iteració
    for it in range(iterations+1): # +1 per incloure l'últim: range(N)=[0, 1, 2...N-1] ; coses de Python...

        field += A * (1-R) * R**it * np.exp(1j*it*delta) # calculem el terme en concret i l'afegim al camp. Això és la suma (in) finita numérica

    if iterations < 8 or it%max(2, iterations//6) == 1: # No dibuixem tots els casos si hi ha moltes iteracions (últim apartat)
        plt.plot(theta_1*180/np.pi, np.abs(field*field.conj()), label=f"N={it}")
```

```
In [32]: # Component ortogonal
plt.title("Orthogonal component")
plot_iterations(Rs, 5)
plt.plot(theta_1*180/np.pi, ITs_geo, 'k', label='analític') # k: color negre
plt.xlabel(r"Incidence angle $\theta_1$ (degrees)")
plt.ylabel(r"Transmitted irradiance $I_{T\perp}/I_0$ (a.u.)")
plt.plot([theta_B, theta_B], [0, 1], '--', label="Brewster angle")
plt.axis([0, 90, 0, 1])
plt.legend(loc='lower left')
```

Out[32]: <matplotlib.legend.Legend at 0x1c6c9ba7d00>

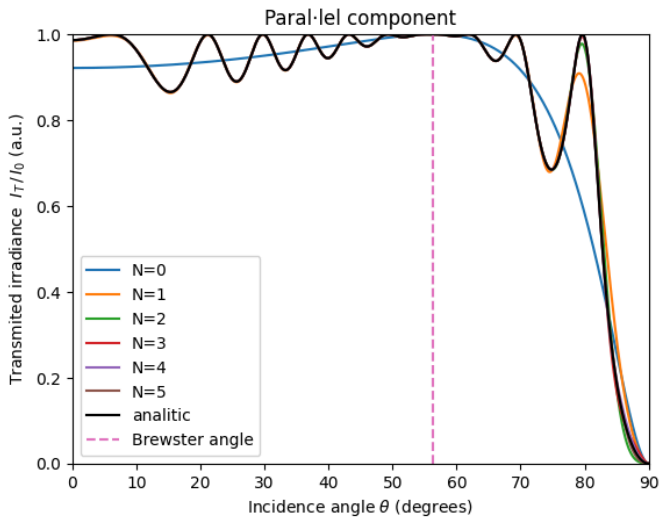


Possible comentari: Veiem que, per angles petits, amb pocs termes (2 o 3) n'hi ha prou per reproduir el resultat analític. En canvi, per a angles més grans calen alguns més (5, 6...). Això és degut a que la reflectivitat depèn de l'angle. A més reflectivitat, més reflexions internes i, per tant, més termes s'han de tenir en compte. Per altra banda, si la reflectivitat és baixa, de seguida la llum surt i les múltiples reflexions internes perden rellevància.

Extra extra! Component paral·lela

```
In [12]: # Component paral·lela
plt.title("Paral·lel component")
plot_iterations(Rp, 5)
plt.plot(theta_1*180/np.pi, ITp_geo, 'k', label='analitic') # k: color negre
plt.plot([theta_B, theta_B], [0, 1], '--', label="Brewster angle")
plt.xlabel(r"Incidence angle $\theta$ (degrees)")
plt.ylabel(r"Transmitted irradiance $I_T/\backslash I_0$ (a.u.)")
plt.axis([0, 90, 0, 1])
plt.legend(loc='lower left')
```

Out[12]: <matplotlib.legend.Legend at 0x1c6c7511ca0>



Possible comentari: Pel cas de la component paral·lela, no calen mai massa termes. Amb dos o tres n'hi ha prou, degut a que la reflectivitat sempre és baixa. Inicialment, perquè en angles petits sempre ho és i, després, perquè prop de l'angle de Brewster també, fent que no creixi massa en cap cas.

Fabry-Perot amb R=0.8

```
In [10]: # Fabry-Perot amb R=0.8

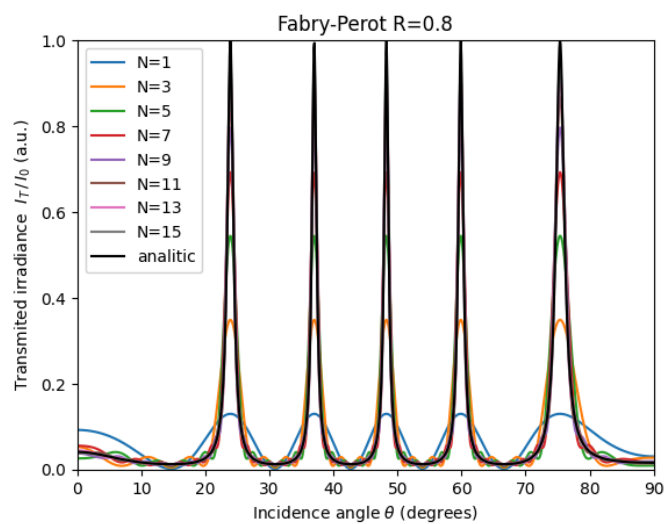
Rfp = 0.8 # en el cas del Fabry-Perot considerem una R constant a 0.8

# Tota la resta és idèntic

ITfp_geo = A**2 / ( 1 + 4*Rfp/(1-Rfp)**2 * np.sin(delta/2)**2 )

plt.title(f"Fabry-Perot R={Rfp}")
plot_iterations(Rfp, 15)
plt.plot(theta_1*180/np.pi, ITfp_geo, 'k', label='analitic') # k: color negre
plt.xlabel(r"Incidence angle $\theta$ (degrees)")
plt.ylabel(r"Transmitted irradiance $I_T/\backslash I_0$ (a.u.)")
plt.axis([0, 90, 0, 1])
plt.legend(loc='upper left')
```

Out[10]: <matplotlib.legend.Legend at 0x13b961214f0>



Possible comentari: En aquest cas la reflectivitat és molt alta, per tant, les reflexions múltiples seran més rellevants. Així que s'ha de tenir en compte més termes. Tot i així, amb 10 o 15 termes és suficient.