Designing Meta-Materials

Jose Luis Del Valle Jr, supervised by David Nicholls
University of Illinois at Chicago

Mathematical Computing Laboratory

UIC

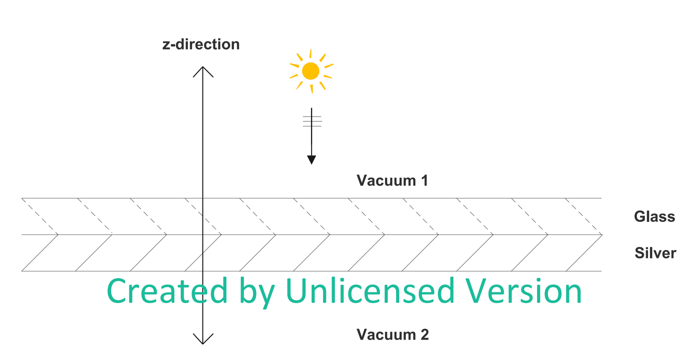
Summary

Meta-materials are assemblies of naturally occurring substances which exhibit unusual properties such as zero permeability and/or permittivity, or negative index of refraction. Applications include high sensitivity diagnostics, super-resolution imaging, and cloaking.

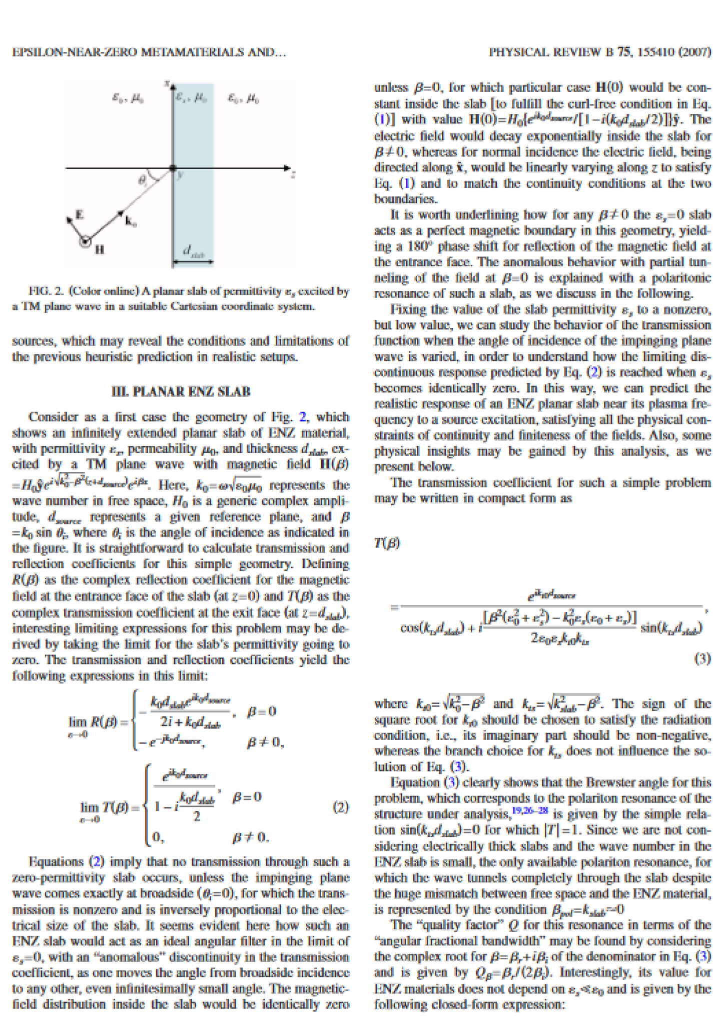
Purpose/Goal

Create a Python code that would serve for an open sourced Machine Learning platform TensorFlow. The Python code solves for the Reflective and Transmittance values. Using this and creating a large database of Transmittance values this would allow the Machine Learning system to optimize the best multi-layered material using the users given parameters of: number of materials, type of material, and angle/strength of wavelength projected onto the meta-surface.

Optics Diagram



About ENZ Materials



EPSILON-NEAR-ZERO METAMATERIALS AND...

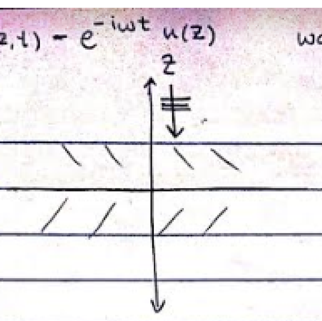
PHYSICAL REVIEW B 75, 155410 (2007)

III. PLANAR ENZ SLAB

Math required

$\underline{u}(z,t) = e^{-i\omega t} u(z)$ wave eqn $\rightarrow$ $u'' + k^2 u = 0$. $k = \omega/c$. (1)

9/9/19



$u \rightarrow$ total field

input

Upper layer $u^{total} = u^{inc} + u^{reflected}$

Lower layer $u^{total} = u^{transmitted}$

scattered.

We want $u^{reflected}$ & $u^{transmitted}$

given $u^{incident}$.

Normal incidence ($\theta_{inc} = 0$)

$u^{incident}(z) = e^{-ik_u z}$ k_u : k in upper layer

$\underline{u}^{incident} = e^{-i\omega t} e^{-ik_u z} = e^{-ik_u(z + \frac{\omega}{k_u} t)}$. ($k_u = \omega / c_u$

c_u.

$= 1 e^{-ik_u(z + c_u t)}$. Sinusoid prop down (if $c_u > 0$) at speed c_u.

Note: $(u^{inc})'' + k_u^2 u^{inc} = 0$. ✓

Seek: $u^{refl}(z)$ & $u^{trans}(z)$ satisfying Helmholtz & B.C.s

Def: $u(z) = u^{refl}(z)$ $z > 0$.

$w(z) = u^{trans}(z)$ $z < 0$

BCs: (1) u^{total} is continuous.

$\rightarrow$ (2) $\begin{cases} \partial_z u^{total} \text{ is cont} & \text{" Transverse Electric Polarization " (TE)} \\ \frac{1}{\sqrt{\epsilon}} \partial_z u^{total} \text{ is cont} & \text{" Magnetic " (TM)} \end{cases}$

(1) $\Rightarrow$ $u^{inc} + u = w$ at $z = 0$.

(2) $\Rightarrow$ $\partial_z u^{inc} + \partial_z u = \partial_z w$ at $z = 0$ $\partial_z = \frac{\partial}{\partial z}$ $\frac{d}{dz}$

$(u^{inc})' + u' = w'$ at $z = 0$.

(3) u is "upward"

(4) w is "downward"

Math required (cont)

(2)

9/9/19.

Problem: $\begin{cases} u'' + k_u^2 u = 0 & z > 0 & ✓ \\ w'' + k_w^2 w = 0 & z < 0 & ✓ \\ u - w = -u^{inc} & \text{at } z = 0. & \text{(iii)} \\ u' - w' = -(u^{inc})' & \text{at } z = 0. & \text{(iv)} \\ u \text{ is upward} & z \to \infty & ✓ \\ w \text{ is downward} & z \to -\infty. & ✓ \end{cases}$

Focus on u: $u = e^{rz} \rightarrow r^2 + k_u^2 = 0. \Rightarrow r = \pm i k_u$

$u = R e^{ik_u z} + A e^{-ik_u z}$.

Note: $\underline{u}(z,t) = e^{-i\omega t} u(z) = R e^{ik_u(z - c_u t)} + A e^{-ik_u(z + c_u t)}$ ω / k_u

up down

Choose $A = 0$. $\boxed{u = R e^{ik_u z}}$

Similarly for w: $w = R e^{ik_w z} + T e^{-ik_w z}$

$\underline{w}(z,t) = e^{-i\omega t} w(z) = R e^{ik_w(z - c_w t)} + T e^{-ik_w(z + c_w t)}$

up down

Choose $R = 0$. $\boxed{w = T e^{-ik_w z}}$

We solve for R & T $\boxed{u^{inc} = e^{-ik_u z}}$

(iii) $\Rightarrow$ $R - T = -1$.

(iv) $\Rightarrow$ $ik_u R - (-ik_w T) = -(ik_u)$ $u' = ik_u R e^{ik_u z}$

$\hookrightarrow k_u R + k_w T = +k_u$ $w' = -ik_w T e^{ik_w z}$

$\begin{pmatrix} 1 & -1 \\ k_u & k_w \end{pmatrix} \begin{pmatrix} R \\ T \end{pmatrix} = \begin{pmatrix} -1 \\ k_u \end{pmatrix}$ $(u^{inc})' = -ik_u e^{-ik_u z}$

Fresnel eqns (R,T) Fresnel coeffs.

Improvements

Some improvements that can be made within this project are:

- Implement a loop or recursive function to solve for many T-Values for the 4 Layer code at once.
- Create a database of many combinations of Meta Materials.
- Create a Multi-Layered code that solves for any number of Layers.

Conclusions

Engineers and others interested in Meta-materials may be able to use this specific 4 Layer code to create a layered structure such as the example provided (Glass/Silver) to find the best proportion of materials that will allow light to transmit through. Thus being able to produce an ENZ-like material.

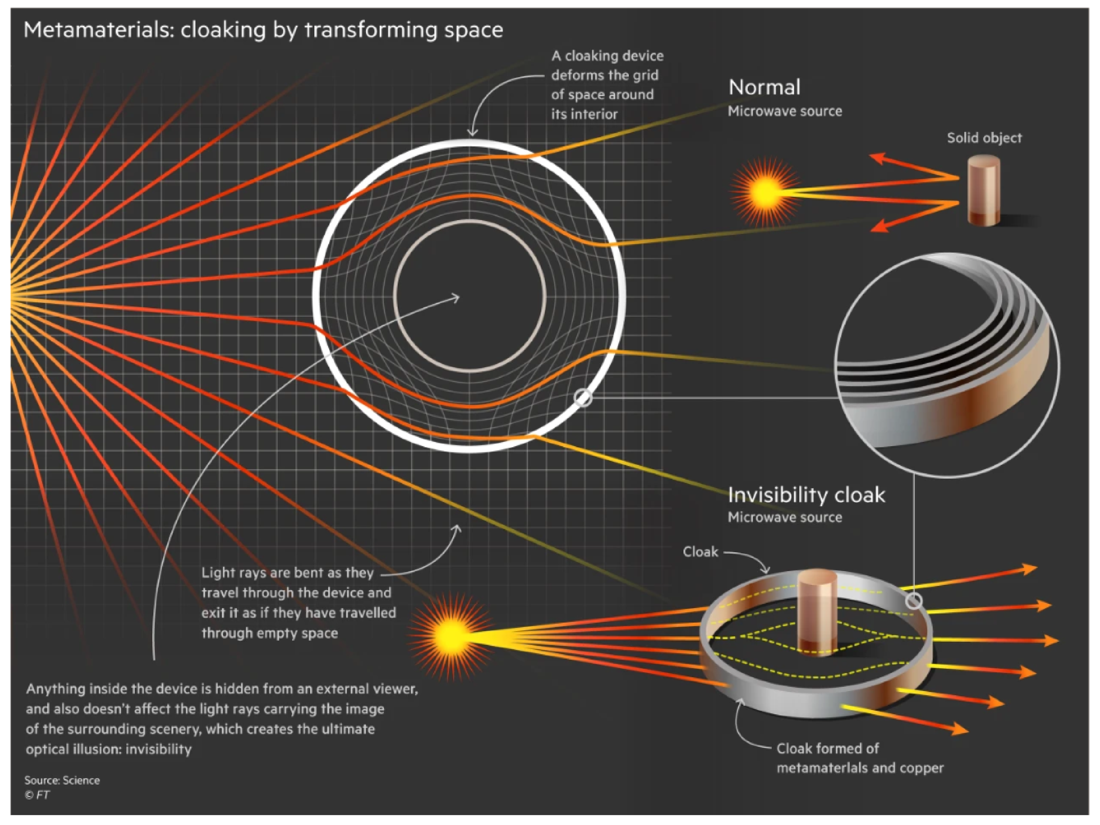
References

https://www.ft.com/content/c6864c76-de7d-11e7-a0d4-0944c5f49e46

https://refractiveindex.info/

Andrea and Silveirinha,Epsilon-near-zero metamaterials and electromagnetic sources: Tailoring the radiation phase pattern. PhysRevB.75.155410 (2007) volume 75,issue 15

Super Cloaking Visual



4 Layered Code:

```python
from numpy import linalg
import numpy as np
from math import *
from cmath import *
import matplotlib.pyplot as plt
```

```python
def TM_four_layer_solver():
        lmda = 0.700            #wavelength
        h = 0.1/2*1.0          #1.0 is 100% of top layer(glass)
        k0 = 2*pi/lmda
        ku = 1*k0
        kv1 = cmath.sqrt(-2.2)*k0 # Permativity of Silver is about -23
        kv2 = cmath.sqrt(-23)*k0 # Permativity of Glass is about 2.2
        kw = cmath.sqrt(1)*k0

        a1 = 0
        a2 = -h/2
        a3 = -h

        tao1 = (ku**2)/(kv1**2)
        tao2 = (kv1**2)/(kv2**2)
        tao3 = (kv2**2)/(kw**2)

        Cu = cos(ku*a1)
        Su = sin(ku*a1)

        Cv1a1 = cos(kv1*a1)
        Sv1a1 = sin(kv1*a1)
        Cv1a2 = cos(kv1*a2)
        Sv1a2 = sin(kv1*a2)

        Cv2a2 = cos(kv2*a2)
        Sv2a2 = sin(kv2*a2)
        Cv2a3 = cos(kv2*a3)
        Sv2a3 = sin(kv2*a3)
```

```python

        Cw = cos(kw*a3)
        Sw = sin(kw*a3)

        nrow1 = [ complex(Cu,Su), complex(-Cv1a1,Sv1a1),complex(-Cv1a1,-Sv1a1), 0, 0, 0 ]
        nrow2 = [ ku*(complex(Cu,Su)), tao1*kv1*(complex(Cv1a1,-Sv1a1)),
                  tao1*kv1*(complex(-Cv1a1,-Sv1a1)), 0, 0, 0 ]
        nrow3 = [ 0, complex(Cv1a2,-Sv1a2), complex(Cv1a2,Sv1a2),
                  complex(-Cv2a2,Sv2a2), complex(-Cv2a2,-Sv2a2), 0 ]
        nrow4 = [ 0, kv1*complex(-Cv1a2,Sv1a2), kv1*complex(Cv1a2,Sv1a2),
                  tao2*kv2*complex(Cv2a2,-Sv2a2), tao2*kv2*complex(-Cv2a2,-Sv2a2), 0 ]
        nrow5 = [ 0, 0, 0, complex(Cv2a3,-Sv2a3), complex(Cv2a3,Sv2a3), complex(-Cw,Sw) ]
        nrow6 = [ 0, 0, 0, kv2*complex(-Cv2a3,Sv2a3), kv2*complex(Cv2a3,Sv2a3), tao3*kw*complex(Cw,-Sw) ]


        nmat = np.array([nrow1,nrow2,nrow3,nrow4,nrow5,nrow6])
        cons = np.array([complex(-Cu,Su), ku*(complex(Cu,-Su)), 0, 0, 0, 0 ])


        answer = np.linalg.solve(nmat,cons)
        Rval = answer[0]
        D1val = answer[1]
        U1val = answer[2]
        D2val = answer[3]
        U2val = answer[4]
        Tval = answer[5]

        print("Reflectance and Transmittance Values: ")
        print("R = ",Rval)
        print("T = ",Tval)
        #print("D1 = ",D1val)
        #print("U1 = ",U1val)
        #print("D2 = ",D2val)
        #print("U2 = ",U2val)

```

4 Layered Code: (cont)

```python
        #Conservation of Energy
        eu = (abs(Rval))**2
        ew = (kw/ku)*(abs(Tval)**2)

        delta = 1 - eu - (ku**2/kw**2)*ew
        print("\n")
        print("Conservation of Energy: ")
        print("delta= ",delta)

def graph_plot():
        P1 = [0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0]
        T1 = [0.9999999999999996,0.9384076579820084,
              0.7905907703317033,0.6231082548491449,
              0.47667332326623363,0.3620531297902782,
              0.2760763820156858,0.2122541242251262,
              0.16468021311102665,0.12885665780945607
              ,0.10155709411010241]
        plt.plot(P1,T1, label = "2 Layered Meta-material")

        P2 = [0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0]
        T2 = [0.999999999996626,0.9994965337635457,
              0.9994965337635457,0.9954748751107413,
              0.9919647447500362,0.9874637502172039,
              0.9819808004934081,0.9755266615560602,
              0.9681138871142562,0.9597567408200651,
              0.9504711112950164]
        plt.plot(P2,T2, label ="ENZ Material")


        plt.xlabel('Percentage')
        plt.ylabel('Transmittance')
        plt.title('Silver/Glass vs ENZ Material')
        plt.legend()
        plt.show()
```

```python

def main():
    TM_four_layer_solver()
    graph_plot()

main()
```

```
Reflectance and Transmittance Values:
R =  (0.6774420628343364+0.6958881648197852j)
T =  (0.045408259970972496-0.23398718659986248j)


Conservation of Energy:
delta=  (2.0816681711721685e-16+0j)
```