

# Trabajo 2 Fotonica

## Librerías y modulos:

Librerias de python a usar:

```
In [1]: import numpy as np
import scipy.special as sp
import scipy.optimize as sp2
import matplotlib.pyplot as plt
import pandas as pd
```

Modulos propios en el documento:

```
In [2]: from core_refractive_index import *
from q1yq2 import *
from TE_TM import *
from Zeros import *
from HE import *
from EH import *
from polar import *
```

## Planteamiento del problema:

Encontrar todos los posibles modos que se pueden propagar en la fibra óptica Corning SMF-28, para una longitud de onda incidente de  $1\mu\text{m}$ . Para el índice de refracción del revestimiento, tomar como material sílice pura ( $\text{SiO}_2$ ) y para el núcleo, sílice dopada con  $\text{GeO}_2$  al 4.2 mol%.

Para obtener todos los datos necesarios para el modelamiento de los modos que se pueden propagar, se hallan los indices de refracción a través de la ecuación de Sellmeier:

$$n(\lambda) = \sqrt{1 + \sum_{i=1}^M A_i \frac{\lambda^2}{\lambda^2 - \lambda_i^2}}$$

Para la realización de estos calculos se uso el articulo To the use of Sellmier formula, que contenía los valores correspondientes a los primeros coeficientes de Sellmier para el  $\text{SiO}_2$  (**calculo en modulo core\_refractive\_index**) y una tabla de silice dopado para  $\text{GeO}_2$  (**Calculo en archivo de matlab**). Y se obtuvieron los siguientes resultados:

```
In [3]: print('Indice de refracción del nucleo: ', n_core())
```

Indice de refracción del nucleo: 1.4568527308462205

```
In [4]: print('Indice de refracción del revestimiento: ', n_cladding())
```

Indice de refracción del revestimiento: 1.450417399464332

Como último dato necesario para el calculo de los modos de propagación de la onda en la fibra óptica, es necesario conocer el radio de la misma, según su hojade datos posee un diametro de  $8.2\mu m$  en el nucleo, por tanto los datos completo serían:

$$a = 4.1\mu m$$

$$n_{co} = 1.45685$$

$$n_{cl} = 1.45042$$

$$\lambda_0 = 1\mu m$$

## Modos TE y TM

Siendo estos indices de refracción muy cercanos, se trata de una situación de guiado debil, por tanto la ecuación de dispersión para los modos TE y TM esta dada por (**contenido en el modulo TE\_TM**):

$$q_1 \frac{J_0(q_1 a)}{J_1(q_1 a)} = - q_2 \frac{K_0(q_2 a)}{K_1(q_2 a)}$$

Donde  $q_1$  y  $q_2$  (**Conteidas en el modulo "q1yq2", con un factor de escala de  $10^6$** ) son respectivamente:

$$q_1 = \sqrt{n_{co}^2 k_0^2 - \beta^2}$$

$$q_2 = \sqrt{\beta^2 - n_{cl}^2 k_0^2}$$

Por tanto:

$$k_0 n_{cl} < \beta < k_0$$

```
In [5]: print(r'El rango de beta es: ( , ko*ncl, ' , ' , ko*nco, ' ) ')
```

El rango de beta es: ( 9113241.293591917 , 9153675.67317743 )

Sabiendo que los ceros de la función se encuentran en la resta de las funciones:

$$f(\beta) = q_1 \frac{J_0(q_1 a)}{J_1(q_1 a)}$$

$$\text{Modo_fibra_optica}$$

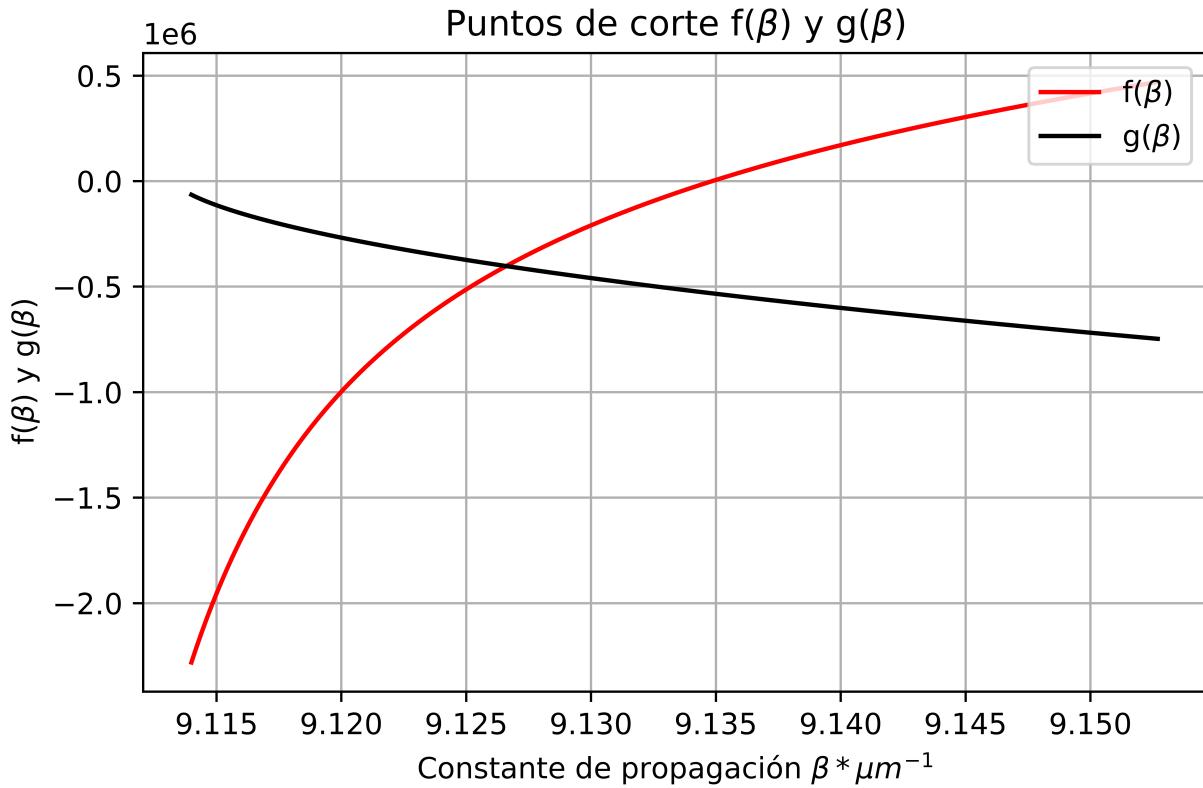
$$K_0(q_2 a)$$

$$g(\beta) = -q_2 \frac{K_0(q_2 a)}{K_1(q_2 a)}$$

Se grafican estas funciones para tener conocimiento del numero modos TE y TM y su ubicación:

```
In [6]: xx = np.linspace(9.114, 9.1527, 500)
yy = tm_te(xx)
yy1 = tm_te_01(xx)
yy2 = tm_te_02(xx)
```

```
In [7]: fig, ax = plt.subplots(dpi=1000)
#ax.plot(xx,yy, label='S(b)', color='black')
ax.plot(xx,yy1, label=r'f($\beta$)', color='red')
ax.plot(xx,yy2, label=r'g($\beta$)', color='black')
ax.set(xlabel=r'Constante de propagación $\beta * \mu m^{-1}$', ylabel=r'f($\beta$) y g($\beta$)')
title=r'Puntos de corte f($\beta$) y g($\beta$)')
ax.grid()
plt.legend(loc='upper right')
fig.tight_layout()
plt.savefig('Puntos de corte TE y TM.png')
plt.show()
```



Solo existe un modo TE y un modo TM, el fundamental, por tanto se puede aplicar el metodo de bisección sobre todo el dominio de la función para hallarlo:

```
In [8]: tm_te_zeros = sp2.bisect(tm_te, 9.114, 9.1527)
neff_TE_TM = (tm_te_zeros / ko) * (10 ** 6)
print('El modo fundamental TE y TM posee una velocidad de propagación de: ', tr
```

El modo fundamental TE y TM posee una velocidad de propagación de: 9.12660823 0777013 con un indice de refracción efectivo: 1.452544813591339

Estonces:

$$TE_{01}: n_{eff} = 1.452544813591339$$

$$TM_{01}: n_{eff} = 1.452544813591339$$

## Modos HE

La ecuación de dispersión para los modos HE en guiado debil (**Contenida en el modulo HE**):

$$q_1 \frac{J_v(q_1 a)}{J_{v-1}(q_1 a)} = q_2 \frac{K_v(q_2 a)}{K_{v-1}(q_2 a)}$$

Por tanto, al graficar las funciones:

$$h(v, \beta) = q_1 \frac{J_v(q_1 a)}{J_{v-1}(q_1 a)}$$

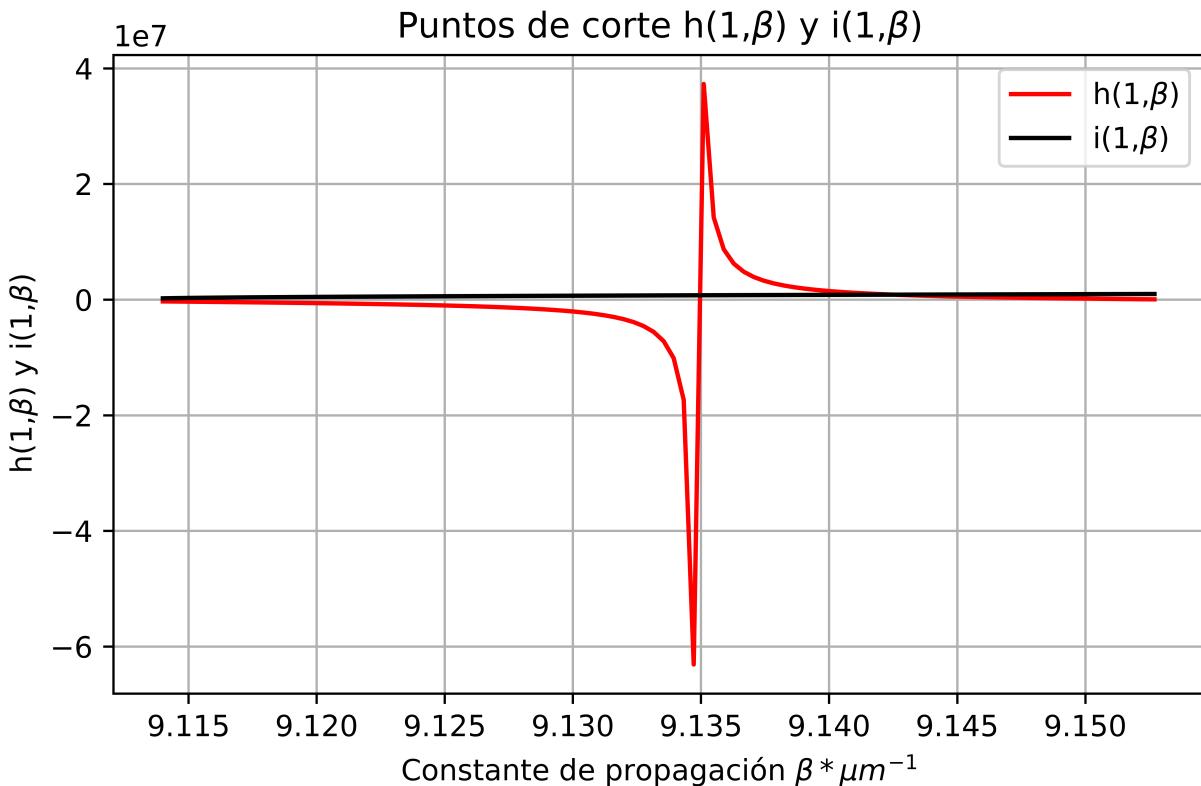
$$i(v, \beta) = q_2 \frac{K_v(q_2 a)}{K_{v-1}(q_2 a)}$$

Podemos saber la ubicación de sus raíces en los puntos de corte, tomando valores discretos de  $v: 1, 2, 3, \dots$

Primeramente graficando los puntos de corte para el modo angular uno:

```
In [9]: xxx = np.linspace(9.114, 9.1527, 100)
yyy = HE(1,xxx)
yyy1 = HE_01(1,xxx)
yyy2 = HE_02(1,xxx)
```

```
In [10]: fig, ax = plt.subplots(dpi=1000)
#ax.plot(xxx,yyy, label='S(b)', color='green')
ax.plot(xxx,yyy1, label=r'h(1,$\beta$)', color='red')
ax.plot(xxx,yyy2, label=r'i(1,$\beta$)', color='black')
ax.set(xlabel=r'Constante de propagación $\beta * \mu^{-1}$', ylabel=r'h(1,$\beta$)')
title=r'Puntos de corte h(1,$\beta$) y i(1,$\beta$)')
ax.grid()
plt.legend(loc='upper right')
fig.tight_layout()
plt.savefig('Puntos de corte HE_v1.png')
plt.show()
```



Se observa que posiblemente hayan dos modos radiales, uno anterior a  $9.130 * 10^6$  y otro posterior a  $9.140 * 10^6$ .

Al evaluar el primer caso, obtenemos que no existe un cambio de signo en la función, ya que en el extremo izquierdo la función toma el valor de:

In [11]: `HE(1, 9.13)`

Out[11]: `-2724002.050815152`

Y en el extremo derecho:

In [12]: `HE(1, 9.113241293591917)`

Out[12]: `-302526.90651117824`

Por tanto, no existe ninguna raíz para valores inferiores a  $9.130 * 10^6$ . Además, entre  $9.130 * 10^6$  y  $9.140 * 10^6$  parece existir un cero, sin embargo, por la forma de la función se puede saber que se trata de una discontinuidad. Por último, entre  $9.140 * 10^6$  y  $9.150 * 10^6$  se puede notar en la grafica que sí existe un cero, por tanto, lo hallamos por el metodo de bisección, que correspondería al modo  $HE_{11}$ .

```
In [13]: def HE_v1(beta):
    HE_v1 = HE(1, beta)
    return HE_v1
```

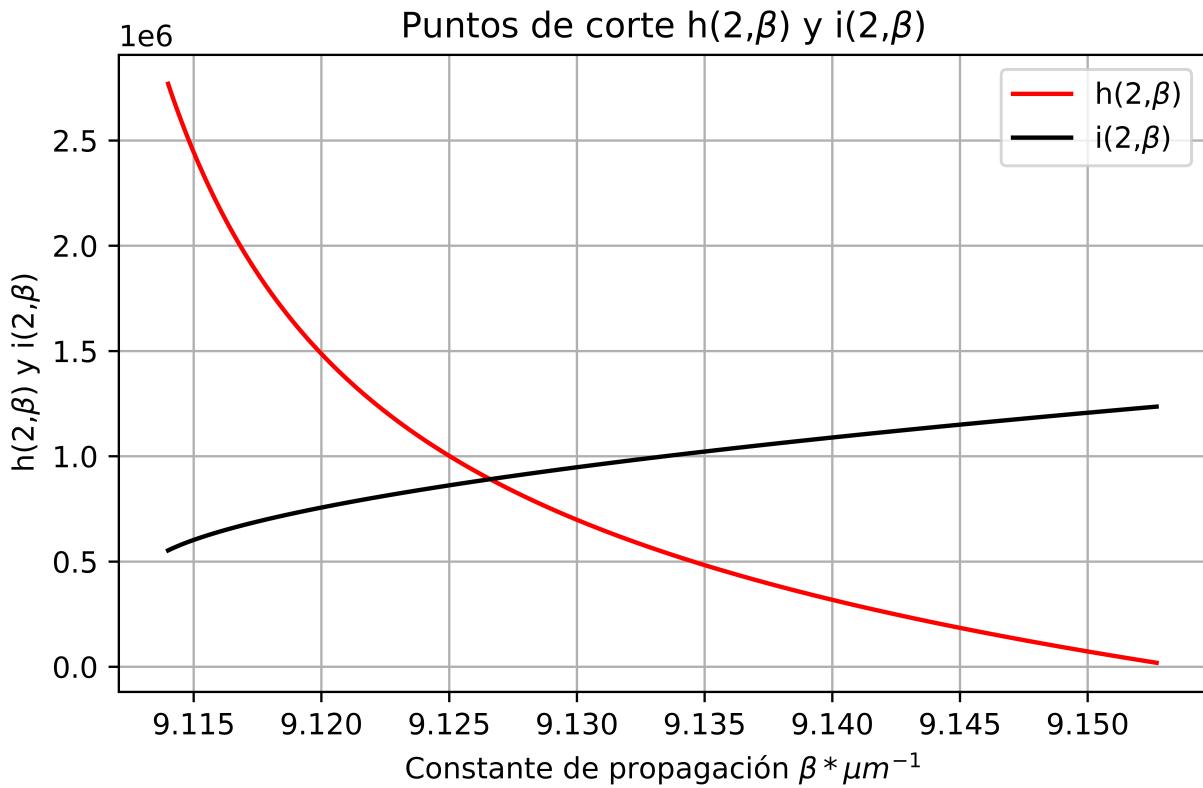
```
HE_zeros_11 = sp2.bisect(HE_v1, 9.140, 9.150)
print('Constante de propagación modo HE11: ', HE_zeros_11)
```

Constante de propagación modo HE11: 9.142541820110056

Realizamos el mismo procedimiento para el modo angular  $v = 2$ :

```
In [14]: xxx_v2 = np.linspace(9.114, 9.1527, 500)
yyy_v2 = HE(2,xxx_v2)
yyy1_v2 = HE_01(2,xxx_v2)
yyy2_v2 = HE_02(2,xxx_v2)
```

```
In [15]: fig, ax = plt.subplots(dpi=1000)
#ax.plot(xxx,yyy, label='S(b)', color='green')
ax.plot(xxx_v2,yyy1_v2, label=r'h(2,$\beta$)', color='red')
ax.plot(xxx_v2,yyy2_v2, label=r'i(2,$\beta$)', color='black')
ax.set(xlabel=r'Constante de propagación $\beta * \mu m^{-1}$', ylabel=r'h(2,$\beta$)')
title=r'Puntos de corte h(2,$\beta$) y i(2,$\beta$)'
ax.grid()
plt.legend(loc='upper right')
fig.tight_layout()
plt.savefig('Puntos de corte HE_v2.png')
plt.show()
```



Se puede observar claramente una raíz, correspondiente al modo  $HE_{21}$ . Hallando dicho cero:

```
In [16]: def HE_v2(beta):
    HE_v2 = HE(2,beta)
    return HE_v2

HE_zeros_21 = sp2.bisect(HE_v2, 9.114, 9.150)
print('Velocidad de propagación modo HE21: ', HE_zeros_21)
```

Velocidad de propagación modo HE21: 9.12660823077767

```
In [17]: neff_HE_11 = (HE_zeros_11 / ko) * (10 ** 6)
neff_HE_21 = (HE_zeros_21 / ko) * (10 ** 6)
print('El modo de propagación HE11 posee una velocidad de propagación de: ', HE
```

El modo de propagación HE11 posee una velocidad de propagación de: 9.142541820110056 y el modo de propagación HE21 posee una velocidad de propagación de: 9.12660823077767 con un indices de refracción efectivo respectivamente: 1.455080723094889 y 1.4525448135914436

Estonces:

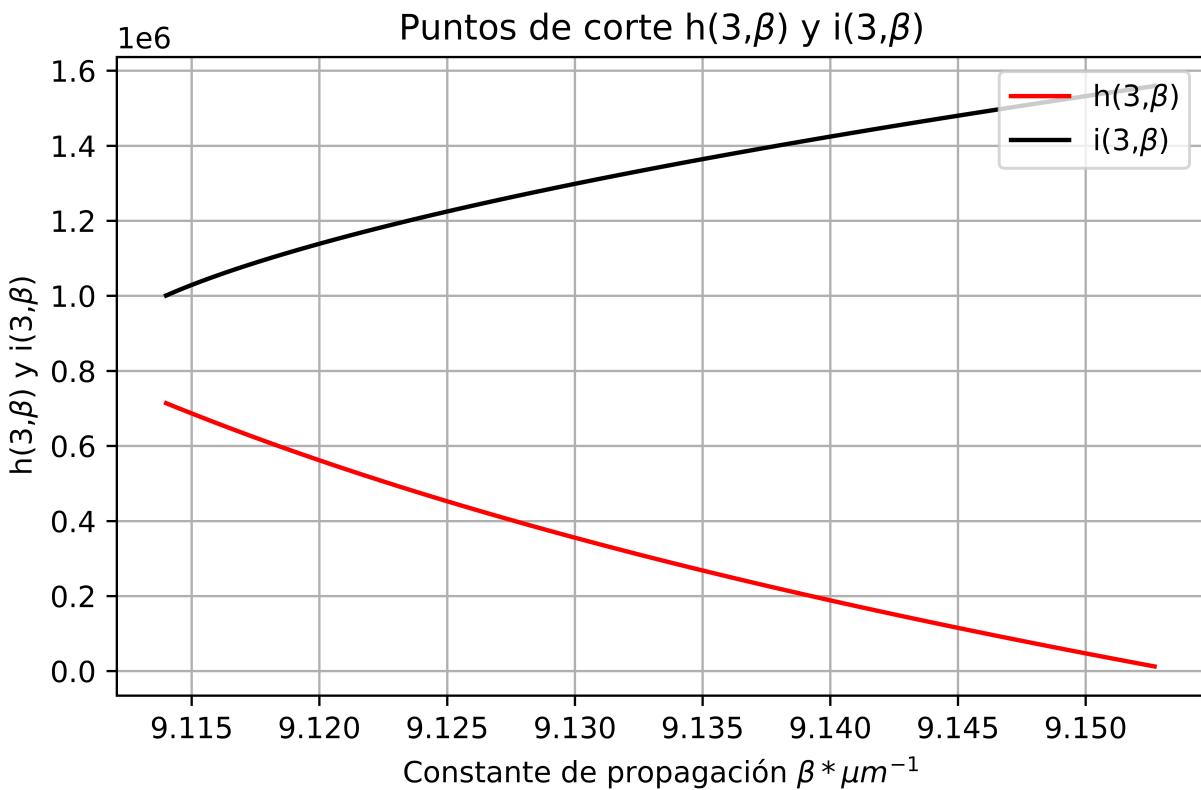
$$HE_{11}: n_{eff} = 1.455080723094889$$

$$HE_{21}: n_{eff} = 1.4525448135914436$$

Volvemos a realizar el mismo procedimiento para el modo angular  $v = 3$ :

```
In [18]: xxx_v3 = np.linspace(9.114, 9.1527, 500)
yyy_v3 = HE(3,xxx_v3)
yyy1_v3 = HE_01(3,xxx_v3)
yyy2_v3 = HE_02(3,xxx_v3)
```

```
In [19]: fig, ax = plt.subplots(dpi=1000)
#ax.plot(xxx,yyy, label='S(b)', color='green')
ax.plot(xxx_v3,yyy1_v3, label=r'h(3,$\beta$)', color='red')
ax.plot(xxx_v3,yyy2_v3, label=r'i(3,$\beta$)', color='black')
ax.set(xlabel=r'Constante de propagación $\beta * \mu m^{-1}$', ylabel=r'h(3,$\beta$)')
title=r'Puntos de corte h(3,$\beta$) y i(3,$\beta$)')
ax.grid()
plt.legend(loc='upper right')
fig.tight_layout()
plt.savefig('Puntos de corte HE_v3.png')
plt.show()
```



Es notorio que no existe ningún punto de corte, por tanto, ya se calcularon todos los modos de propagación posibles para los modos híbridos HE.

## Modos EH

La ecuación de dispersión para los modos híbridos EH es:

$$q_1 \frac{J_v(q_1 a)}{J_{v+1}(q_1 a)} = -q_2 \frac{K_v(q_2 a)}{K_{v+1}(q_2 a)}$$

Es posible visualizar las raíces en los puntos de intersección de estas dos funciones:

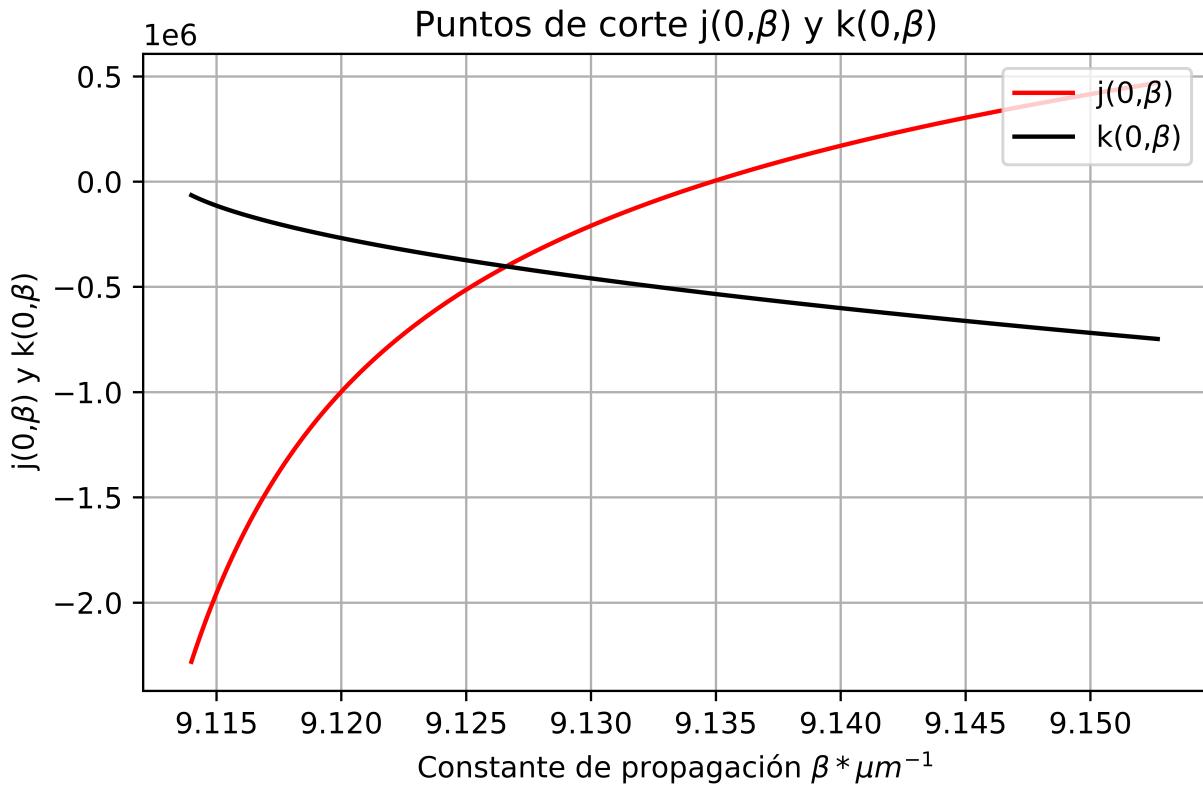
$$j(v, \beta) = q_1 \frac{J_v(q_1 a)}{J_{v+1}(q_1 a)}$$

$$k(v, \beta) = -q_2 \frac{K_v(q_2 a)}{K_{v+1}(q_2 a)}$$

Graficando los puntos de corte para el primer modo angular:

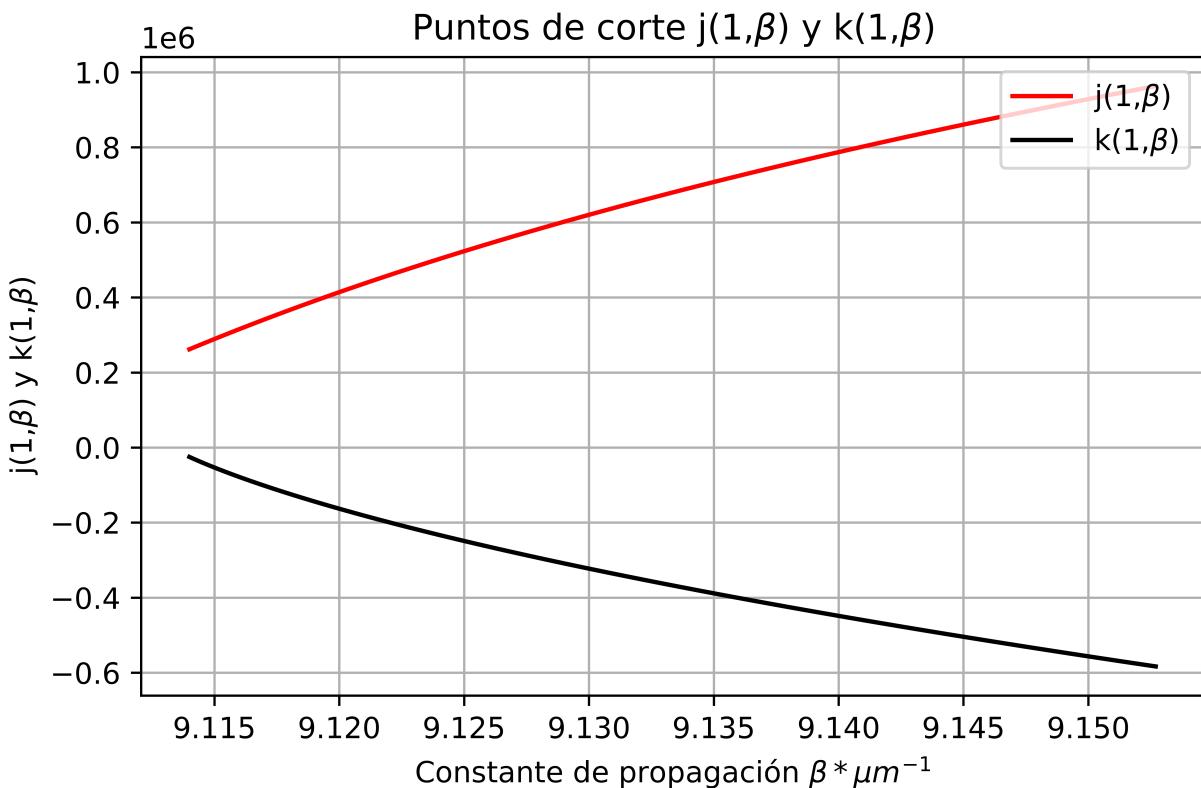
```
In [20]: x_EH_v0 = np.linspace(9.114, 9.1527, 500)
y_EH_v0 = EH(0, x_EH_v0)
y_EH1_v0 = EH_01(0, x_EH_v0)
y_EH2_v0 = EH_02(0, x_EH_v0)
```

```
In [21]: fig, ax = plt.subplots(dpi=1000)
ax.plot(x_EH_v0,y_EH1_v0, label=r'j(0,$\beta$)', color='red')
ax.plot(x_EH_v0,y_EH2_v0, label=r'k(0,$\beta$)', color='black')
ax.set(xlabel=r'Constante de propagación $\beta * \mu m^{-1}$', ylabel=r'j(0,$\beta$)')
title=r'Puntos de corte j(0,$\beta$) y k(0,$\beta$)')
ax.grid()
plt.legend(loc='upper right')
fig.tight_layout()
plt.savefig('Puntos de corte EH_v0.png')
plt.show()
```



```
In [22]: x_EH_v1 = np.linspace(9.114,9.1527,500)
y_EH_v1 = EH(1,x_EH_v1)
y_EH1_v1 = EH_01(1,x_EH_v1)
y_EH2_v1 = EH_02(1,x_EH_v1)
```

```
In [23]: fig, ax = plt.subplots(dpi=1000)
#ax.plot(xxx,yyy, label='S(b)', color='green')
ax.plot(x_EH_v1,y_EH1_v1, label=r'j(1,$\beta$)', color='red')
ax.plot(x_EH_v1,y_EH2_v1, label=r'k(1,$\beta$)', color='black')
ax.set(xlabel=r'Constante de propagación $\beta * \mu m^{-1}$', ylabel=r'j(1,$\beta$)')
title=r'Puntos de corte j(1,$\beta$) y k(1,$\beta$)')
ax.grid()
plt.legend(loc='upper right')
fig.tight_layout()
plt.savefig('Puntos de corte EH_v1.png')
plt.show()
```



Dado que no existe ningún punto de corte, entonces tampoco hay ningún modo de propagación híbrido EH.

Por tanto los modos de propagación presentes en la fibra óptica son:

$$TE_{01}: n_{eff} = 1.452544813591339$$

$$TM_{01}: n_{eff} = 1.452544813591339$$

$$HE_{11}: n_{eff} = 1.455080723094889$$

$$HE_{21}: n_{eff} = 1.4525448135914436$$

## Modelamiento de campos

En esta sección se modelaran los campos  $|\vec{E}|^2$ ,  $H_z$  y  $E_z$ , donde las expresiones correspondientes a estos campos son:

$$E_z = AJ_v(q_1 r) e^{jv\theta} e^{j(wt - \beta z)} \quad r \leq a$$

$$E_z = CK_v(q_2 r) e^{jv\theta} e^{j(wt - \beta z)} \quad r > a$$

Para el campo magnético:

$$H_z = BJ_v(q_1r)e^{j\nu\theta}e^{j(wt-\beta z)} \quad r \leq a$$

$$H_z = DK_v(q_2r)e^{j\nu\theta}e^{j(wt-\beta z)} \quad r > a$$

Donde las constantes A, B, C y D se pueden hallar al resolver el siguiente sistema de ecuaciones:

$$\begin{bmatrix} J_v(q_1a) & 0 & -K_v(q_2a) & 0 \\ \frac{\beta\nu}{q_1^2a}J_v(q_1a) & \frac{j\omega\mu_0}{q_1}J_v(q_1a) & \frac{\beta\nu}{q_2^2a}K_v(q_2a) & \frac{j\omega\mu_0}{q_2}K_v(q_2a) \\ 0 & J_v(q_1a) & 0 & -K_v(q_2a) \\ -\frac{j\omega\epsilon_1}{q_1}J_v(q_1a) & \frac{\beta\nu}{q_1^2a}J_v(q_1a) & -\frac{j\omega\epsilon_2}{q_2}K_v(q_2a) & \frac{\beta\nu}{q_2^2a}K_v(q_2a) \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

De donde se tiene que:

$$\frac{C}{A} = \frac{J_v(q_1a)}{K_v(q_2a)}$$

$$\frac{B}{A} = \frac{i\beta\nu}{\omega\mu_0} \left( \frac{1}{q_2^2a^2} + \frac{1}{q_1^2a^2} \right) \left( \frac{J_v'(q_1a)}{q_1aJ_v(q_1a)} + \frac{K_v'(q_2a)}{q_2aK_v(q_2a)} \right)^{-1}$$

$$\frac{D}{A} = \frac{J_v(q_1a)}{K_v(q_2a)} \frac{B}{A}$$

Para los modos fundamentales, donde el modo angular es cero, se tienen dos posibles soluciones, la primera que  $B = 0$ , por lo que solo sería útil la ecuación que relaciona a C y a A, es decir, la componente 'z' del campo eléctrico, ya que la componente 'z' del campo magnético sería cero, por lo que se estaría tomando el modo  $TM_{0m}$ ; por otro lado, también se puede tener el caso donde  $A = \infty$ , por tanto se estaría trabajando con  $TE_{0m}$ , ya que la componente 'z' del campo eléctrico es nula.

Además, las componentes  $E_r$ ,  $E_\theta$ ,  $H_r$  y  $H_\theta$  se pueden expresar respecto a las componentes eléctricas y magnéticas en 'z' como:

$$E_r = \frac{i}{q_1^2} \left( \beta \frac{\partial E_z}{\partial r} + \mu_0 \frac{\omega}{r} \frac{\partial H_z}{\partial \phi} \right)$$

$$E_\phi = \frac{i}{q_1^2} \left( \beta \frac{\partial E_z}{r \partial \phi} - \mu_0 \omega \frac{\partial H_z}{\partial r} \right)$$

$$H_r = \frac{i}{q_1^2} \left( \beta \frac{\partial H_z}{\partial r} - \epsilon_0 n^2 \frac{\omega}{r} \frac{\partial E_z}{\partial \phi} \right)$$

$$H_\phi = \frac{i}{q_1^2} \left( \beta \frac{\partial H_z}{r \partial \phi} + \epsilon_0 n^2 \omega \frac{\partial E_z}{\partial r} \right)$$

## Modo $TM_{01}$

Teniendo en cuenta la información anterior, para el modo  $TM_{01}$  solo tendría relevancia el campo eléctrico en el eje 'z', por lo que primeramente expresamos la constante  $C$  en términos de  $A$ :

$$C = \frac{J_0(q_1 a)}{K_0(q_2 a)} A$$

Y posteriormente lo reemplazamos en la expresión para el campo eléctrico en dirección 'z', omitiendo la exponencial que representa la propagación temporal y a lo largo del eje 'z', ya que se graficará solamente la sección transversal de la fibra óptica:

$$E_z = A J_0(q_1 r) \quad r \leq a$$

$$E_z = \frac{J_0(q_1 a)}{K_0(q_2 a)} A K_0(q_2 r) \quad r > a$$

Fijando la constante  $A$  como 1, ya que la amplitud de las funciones de Bessel de orden cero es uno y por tanto la amplitud de la componente dominante ( $E_z$ ) será 1, graficamos la parte real del campo:

```
In [24]: a = 4.1 * (10 ** -6)
```

```
def Ez_TM_01_co(r, theta):
    Ez = sp.jv(0, q1(tm_te_zeros)**r)
    Ezi = np.real(Ez)
    return Ezi

def Ez_TM_01_cl(r, theta):
```

```
Ez = (sp.jv(0,q1(tm_te_zeros)*a) / sp.kn(0,q2(tm_te_zeros)*a)) * sp.kn(0,q2(tm_te_zeros)*a)
Ezi = np.real(Ez)
return Ezi
```

In [73]:

```
#core
dr, dtheta = a/500, (np.pi)/500
r, theta = np.mgrid[-a:a+dr:dr, 0:2*np.pi+dtheta:dtheta]

zz = Ez_TM_01_co(r,theta)
zz = zz[::-1, ::-1]
z_min, z_max = zz.min(), zz.max()

#cladding
dr_cl, dtheta_cl = (1*a)/500, (2*np.pi)/500
r_cl, theta_cl = np.mgrid[a:(1.5*a)+dr_cl:dr_cl, 0:2*np.pi+dtheta_cl:dtheta_cl]

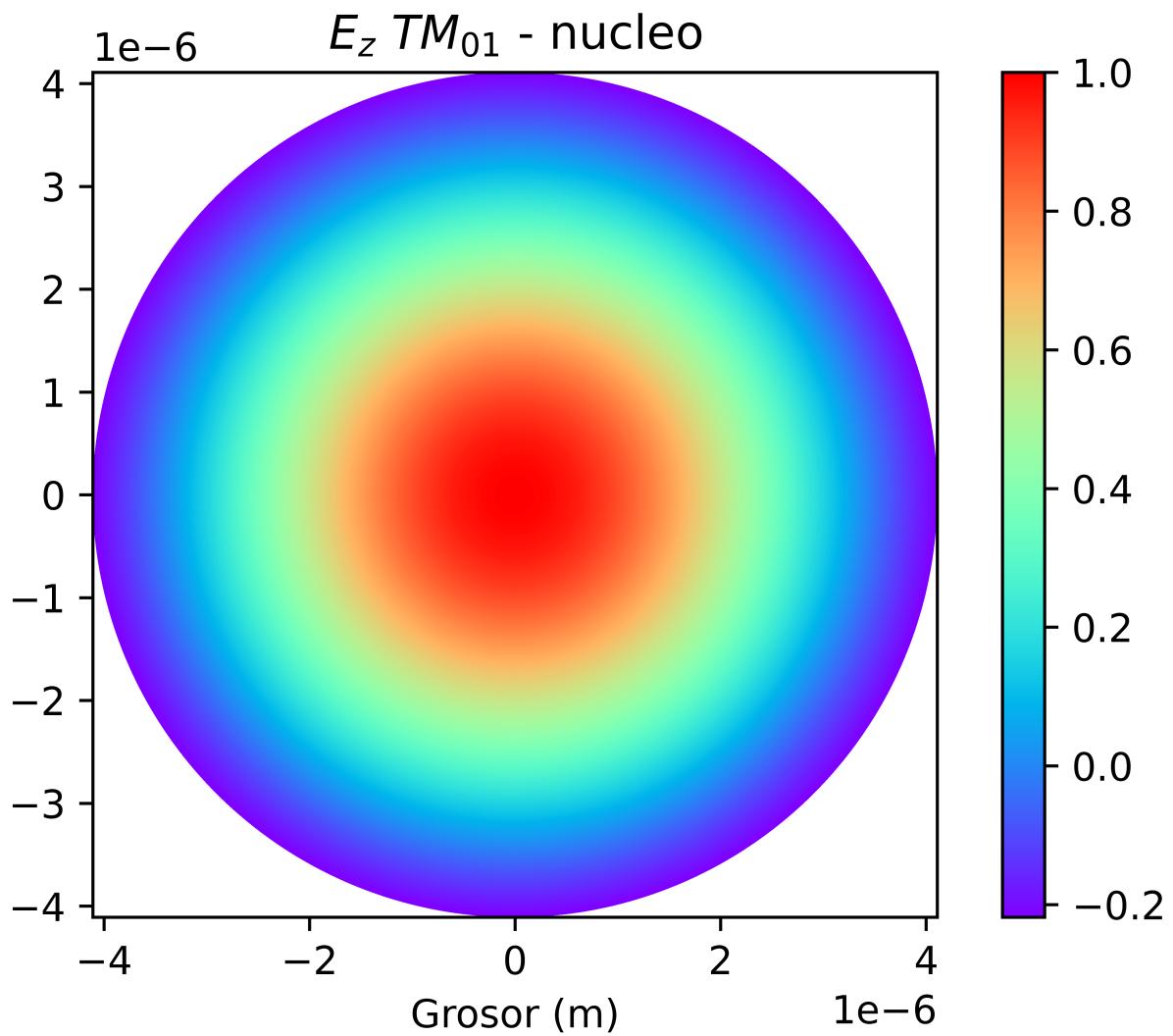
zz_cl = Ez_TM_01_cl(r_cl,theta_cl)
zz_cl = zz_cl[::-1, ::-1]
z_min_cl, z_max_cl = zz_cl.min(), zz_cl.max()

np.shape(np.mgrid[-a:a+dr:dr, 0:2*np.pi+dtheta:dtheta])
```

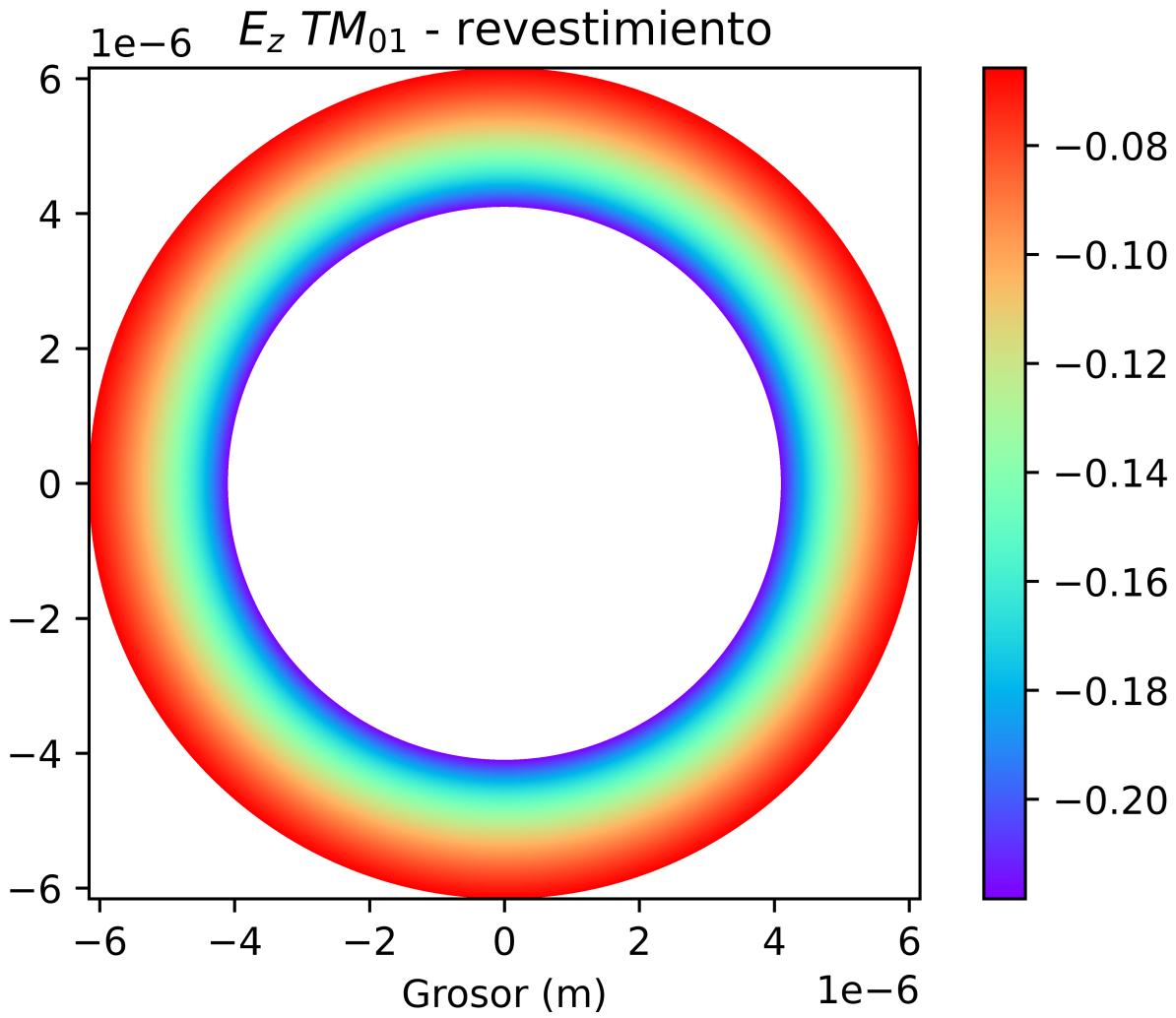
Out[73]: (2, 1002, 1001)

In [70]:

```
fig, axs = plt.subplots(dpi=1000)
ax = axs
axs.set_aspect('equal', adjustable='box')
c1 = ax.pcolor(polar_car(r,theta)[0], polar_car(r,theta)[1], zz, cmap='rainbow')
#ax.set_title('Campo eléctrico en dirección z - perfil transversal')
ax.set(xlabel=r'Grosor (m)',
       title=r'$E_z$ $TM_{01}$ - nucleo')
fig.colorbar(c1, ax=ax)
plt.savefig('Ez_TM01_core.png')
```



```
In [27]: fig, axs = plt.subplots(dpi=1000)
ax = axs
ax.set_aspect('equal', adjustable='box')
c2 = ax.pcolor(polar_car(r_cl,theta_cl)[0], polar_car(r_cl,theta_cl)[1], zz_cl
#ax.set_title('pcolor')
ax.set(xlabel=r'Grosor (m)',
title=r'$E_z$ $TM_{01}$ - revestimiento')
fig.colorbar(c2, ax=ax)
plt.savefig('Ez_TM01_cladding.png')
```



Luego, hallando la norma cuadrada del campo electrico:

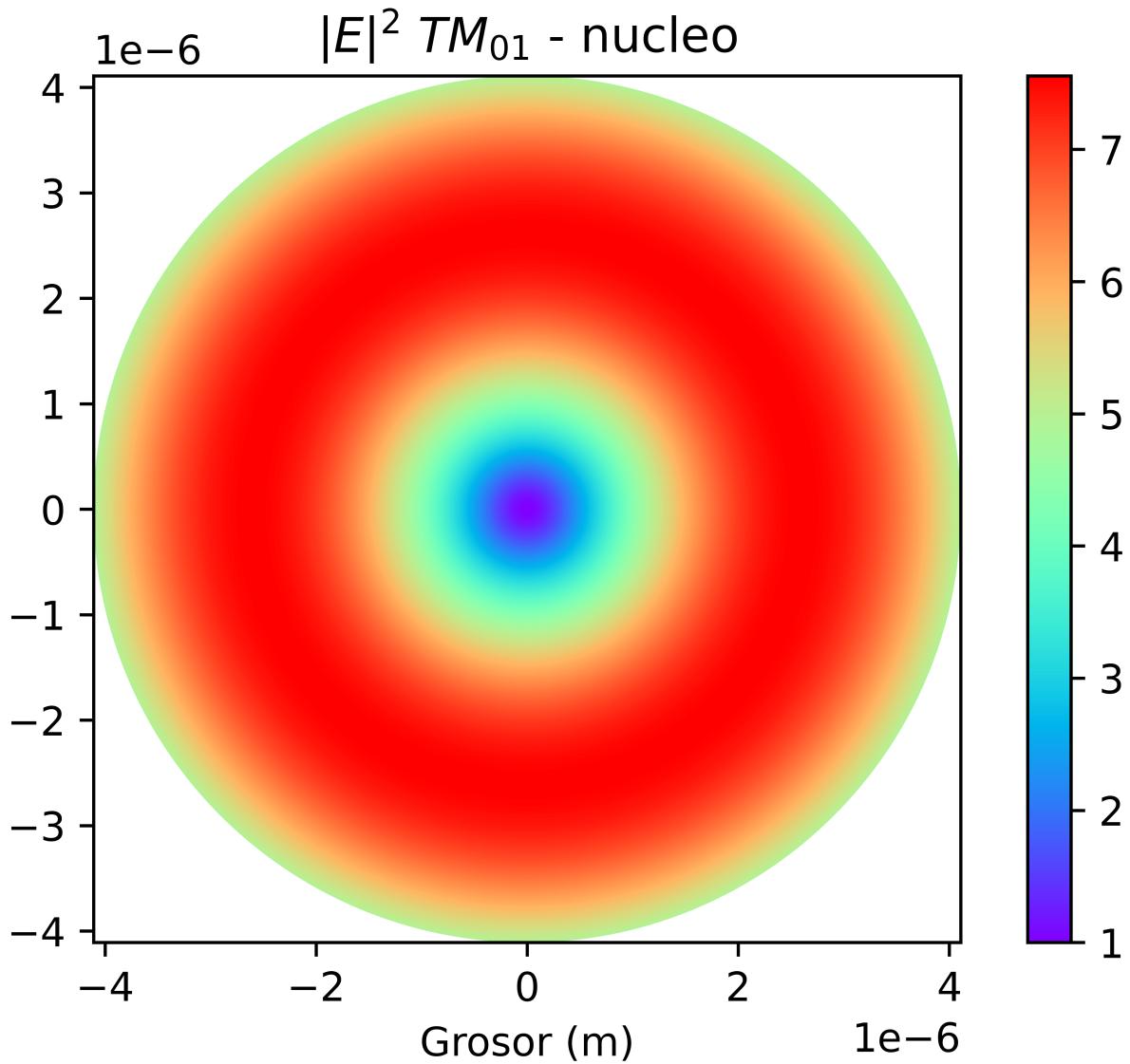
```
In [28]: def normE_co_TM01(r,theta):
    b = tm_te_zeros
    w = (2*np.pi*(3*(10**6)))
    mu = (4*np.pi*(10**-7))
    Ez = sp.jv(0,q1(b)*r)
    Er = -((b*(10**6))*sp.jv(1,q1(b)*r))/q1(b)
    Et = 0
    normE2 = np.sqrt(Ez*np.conj(Ez) + Er*np.conj(Er) + Et*np.conj(Et))
    normEr = np.real(Er)*np.real(Er)
    return normE2
```

```
In [29]: dr_normE_TM01, dtheta_normE_TM01 = a/500, (2*np.pi)/500
r_normE_TM01, theta_normE_TM01 = np.mgrid[-a:a+dr_normE_TM01:dr_normE_TM01, 0:2*np.pi:dtheta_normE_TM01]

zz_normE_TM01 = normE_co_TM01(r_normE_TM01,theta_normE_TM01)
zz_normE_TM01 = np.real(zz_normE_TM01[:-1, :-1])
z_min_normE_TM01, z_max_normE_TM01 = zz_normE_TM01.min(), zz_normE_TM01.max()
```

```
In [30]: fig, axs = plt.subplots(dpi=1000)
ax = axs
ax.set_aspect('equal', adjustable='box')
c1 = ax.pcolor(polar_car(r_normE_TM01,theta_normE_TM01)[0], polar_car(r_normE_TM01,theta_normE_TM01)[1], zz_normE_TM01)
```

```
ax.set(xlabel=r'Grosor (m)',  
       title=r'$|E|^2 TM_{01}$ - nucleo')  
fig.colorbar(c1, ax=ax)  
plt.savefig('NormE_TM01_core.png')
```



## Modo $TE_{01}$

En el caso de los modos  $TE_{01}$  quién tiene relevancia es el campo magnético en dirección z, es decir,  $H_z$ , por tanto, la relación a considerar es:

$$D = \frac{J_0(q_1 a)}{K_0(q_2 a)} B$$

Reemplazando la constante D, escrita respecto a B, y omitiendo la propagación a lo largo de 'z' y la variación temoral, el campo magnético se expresa como:

$$\text{Modo_fibra_optica}$$

$$H_z = BJ_0(q_1 r) \quad r \leq a$$

$$H_z = \frac{J_0(q_1 a)}{K_0(q_2 a)} BK_0(q_2 r) \quad r > a$$

Tomando  $B = 1$  y hallando la componente real del campo  $H_z$ :

```
In [31]: def Hz_TE_01_co(r, theta):
    a = 4.1 * (10 ** -6)
    Hz = sp.jv(0, q1(tm_te_zeros)*r)
    Hzi = np.real(Hz)
    return Hzi

def Hz_TE_01_cl(r, theta):
    a = 4.1 * (10 ** -6)
    Ez = (sp.jv(0, q1(tm_te_zeros)*a) / sp.kn(0, q2(tm_te_zeros)*a)) * sp.kn(0, q2(tm_te_zeros)*r)
    Ezi = np.real(Ez)
    return Ezi
```

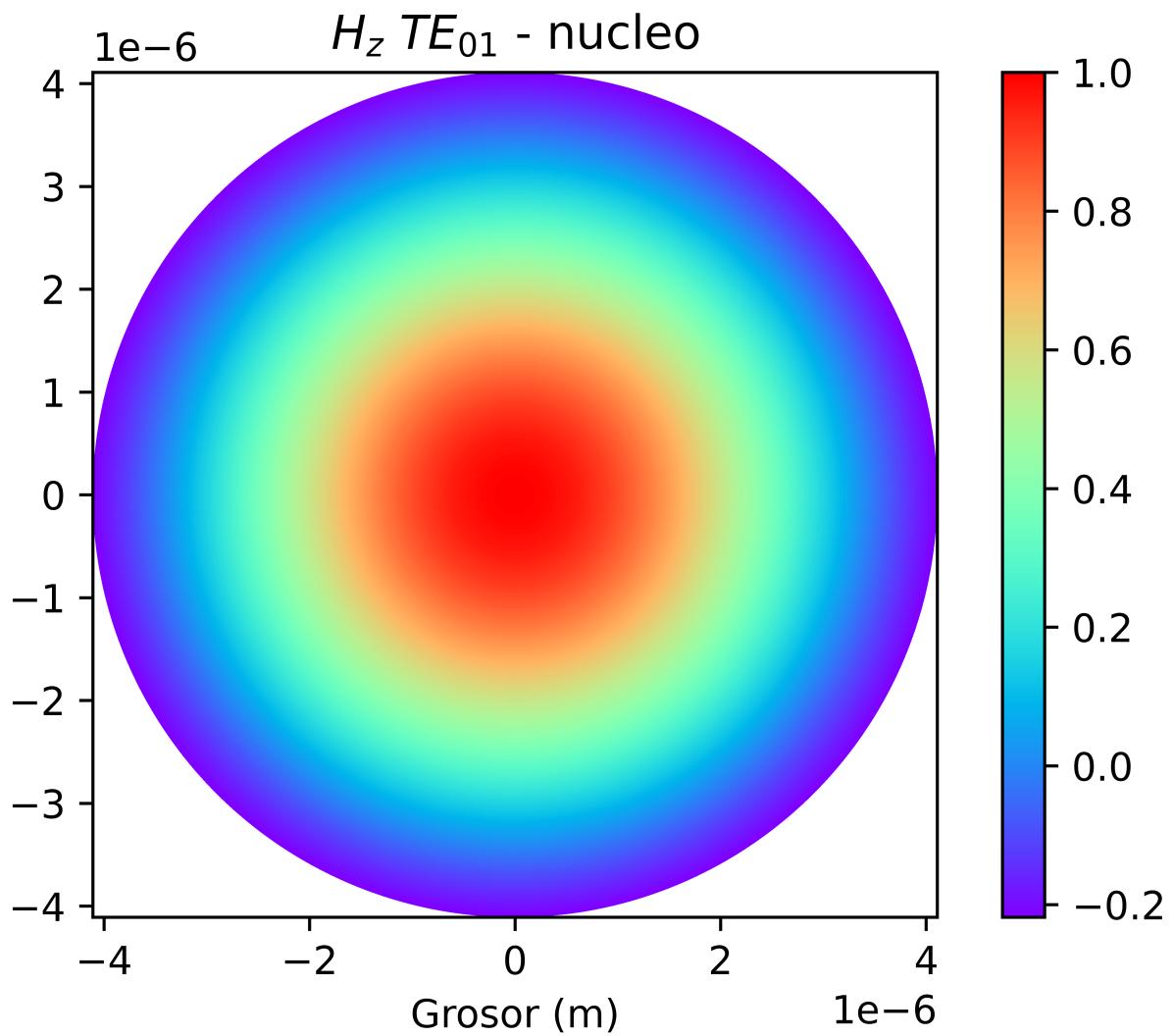
```
In [32]: #core
dr_TE, dtheta_TE = a/500, (2*np.pi)/500
r_TE, theta_TE = np.mgrid[-a:a+dr_TE:dr_TE, 0:2*np.pi+dtheta_TE:dtheta_TE]

zz_TE = Hz_TE_01_co(r_TE, theta_TE)
zz_TE = zz_TE[:-1, :-1]
z_min_TE, z_max_TE = zz_TE.min(), zz_TE.max()

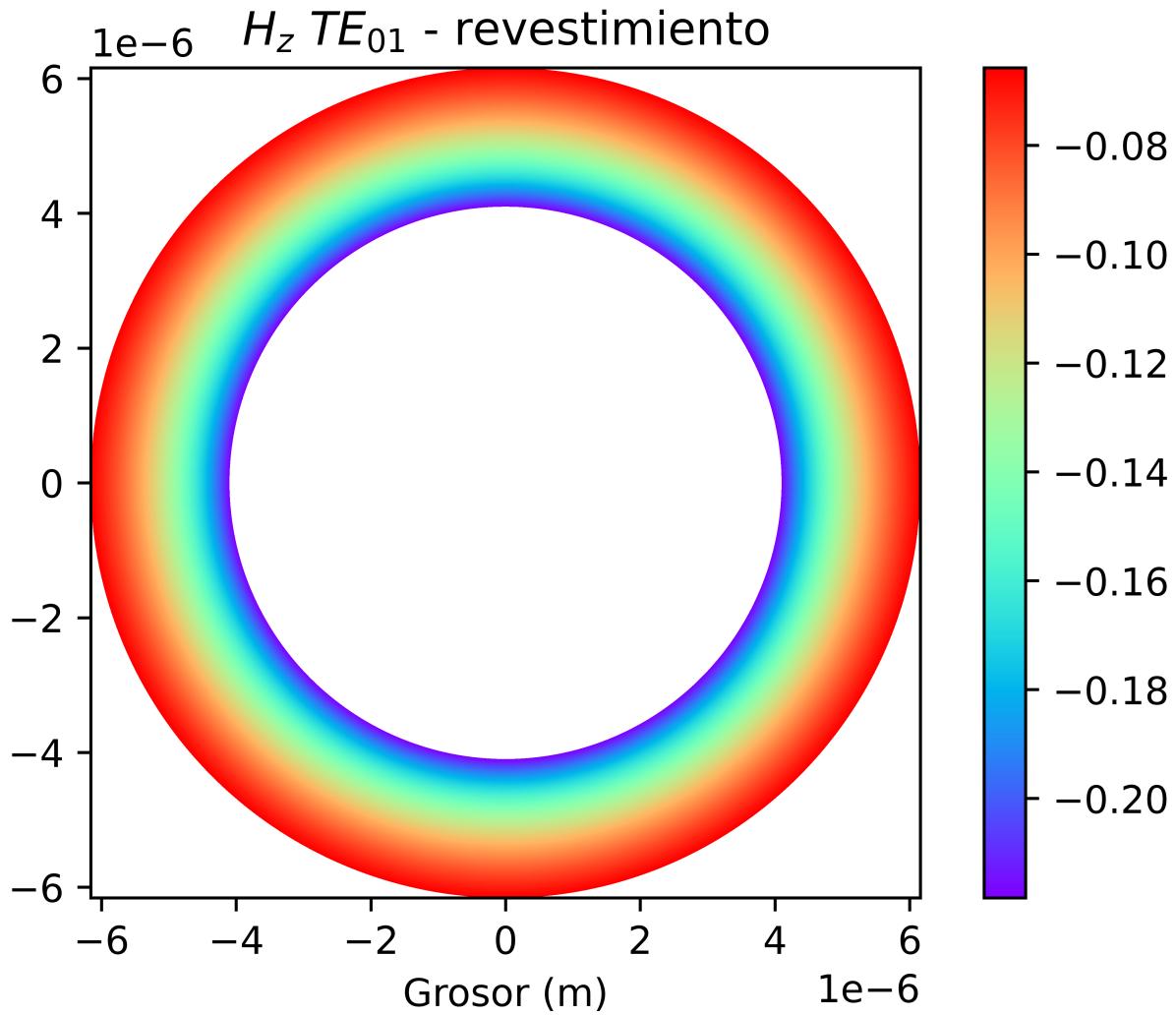
#cladding
dr_cl_TE, dtheta_cl_TE = (1*a)/500, (2*np.pi)/500
r_cl_TE, theta_cl_TE = np.mgrid[a:(1.5*a)+dr_cl_TE:dr_cl_TE, 0:2*np.pi+dtheta_cl_TE:dtheta_cl_TE]

zz_cl_TE = Hz_TE_01_cl(r_cl_TE, theta_cl_TE)
zz_cl_TE = zz_cl_TE[:-1, :-1]
z_min_cl_TE, z_max_cl_TE = zz_cl_TE.min(), zz_cl_TE.max()
```

```
In [33]: fig, axs = plt.subplots(dpi=1000)
ax = axs
axs.set_aspect('equal', adjustable='box')
c1 = ax.pcolor(polar_car(r_TE, theta_TE)[0], polar_car(r_TE, theta_TE)[1], zz_TE)
ax.set(xlabel=r'Grosor (m)',
       title=r'$H_z$ $TE_{01}$ - nucleo')
fig.colorbar(c1, ax=ax)
plt.savefig('Hz_TE01_core.png')
```



```
In [34]: fig, axs = plt.subplots(dpi=1000)
ax = axs
ax.set_aspect('equal', adjustable='box')
c2 = ax.pcolor(polar_car(r_cl_TE,theta_cl_TE)[0], polar_car(r_cl_TE,theta_cl_TE)[1], H_z_TE01_cladding)
#ax.set_title('pcolor')
ax.set(xlabel=r'Grosor (m)',
       title=r'$H_z$ $TE_{01}$ - revestimiento')
fig.colorbar(c2, ax=ax)
plt.savefig('Hz_TE01_cladding.png')
```



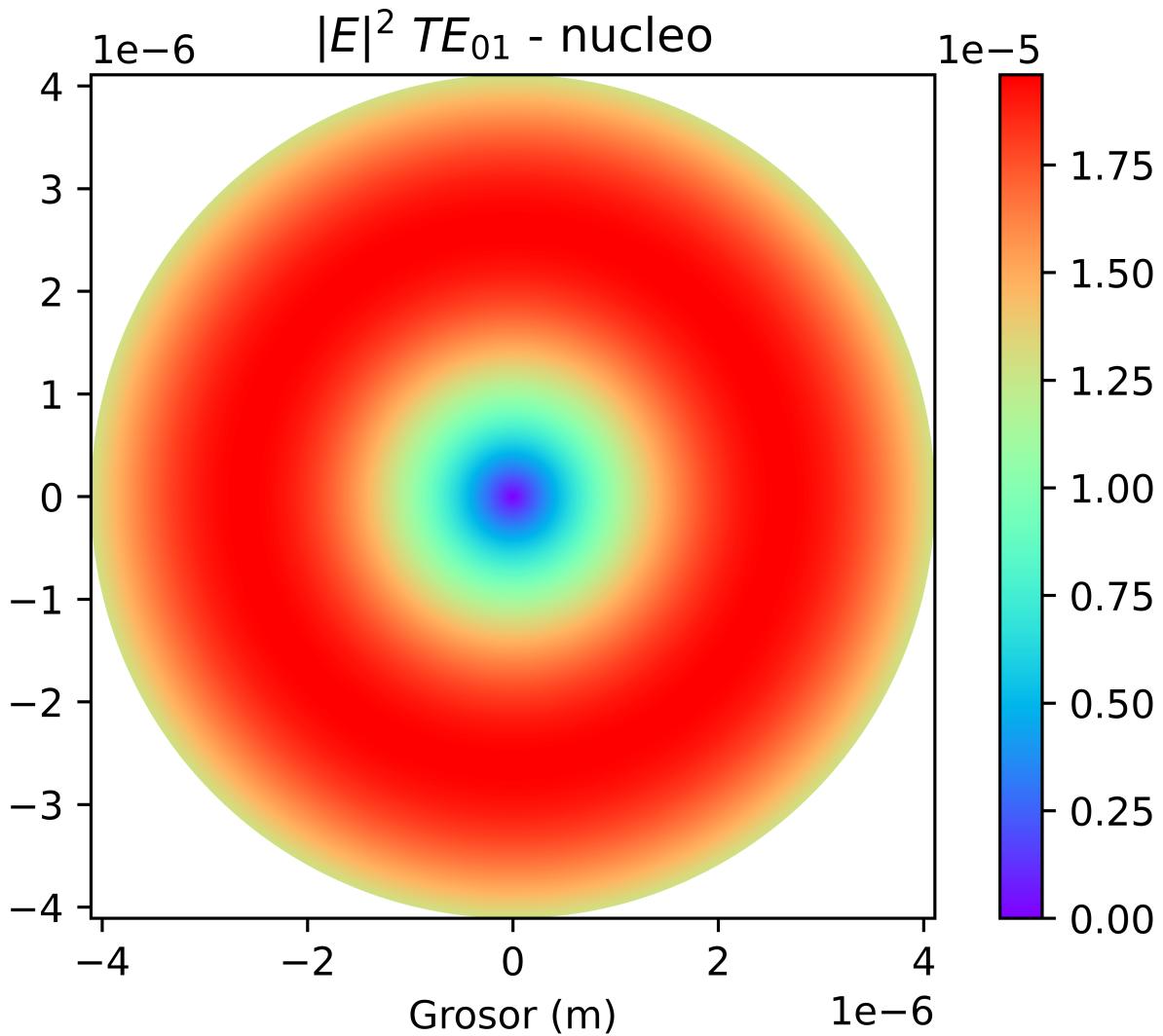
Luego, hallando la norma cuadrada del campo electrico:

```
In [35]: def normE_co_TE01(r,theta):
    b = tm_te_zeros
    w = (2*np.pi* (3 * (10 ** 6)))
    mu = (4*np.pi* (10 ** -7))
    Ez = 0
    Er = 0
    Et = ((1j)*mu*w*sp.jv(1,q1(b)*r))/q1(b)
    normE2 = np.sqrt(Ez*np.conj(Ez) + Er*np.conj(Er) + Et*np.conj(Et))
    return normE2
```

```
In [36]: dr_normE_TE01, dtheta_normE_TE01 = a/500, (2*np.pi)/500
r_normE_TE01, theta_normE_TE01 = np.mgrid[-a:a+dr_normE_TE01:dr_normE_TE01, 0:2*pi]
zz_normE_TE01 = normE_co_TE01(r_normE_TE01,theta_normE_TE01)
zz_normE_TE01 = np.real(zz_normE_TE01[:,-1,:-1])
z_min_normE_TE01, z_max_normE_TE01 = zz_normE_TE01.min(), zz_normE_TE01.max()
```

```
In [37]: fig, axs = plt.subplots(dpi=1000)
ax = axs
ax.set_aspect('equal', adjustable='box')
c1 = ax.pcolor(polar_car(r_normE_TE01,theta_normE_TE01)[0], polar_car(r_normE_TE01,theta_normE_TE01)[1], zz_normE_TE01)
ax.set(xlabel=r'Grosor (m)',
```

```
title=r'$|E|^2 TE_{01} - nucleo$'
fig.colorbar(c1, ax=ax)
plt.savefig('NormE_TE01_core.png')
```



## Modo $HE_{11}$

Las relaciones entre las constantes del modo híbrido  $HE_{11}$  quedarían de la siguiente manera:

$$\frac{C}{A} = \frac{J_1(q_1 a)}{K_1(q_2 a)}$$

$$\frac{B}{A} = \frac{i\beta}{\omega\mu_0} \left( \frac{1}{q_2^2 a^2} + \frac{1}{q_1^2 a^2} \right) \left( \frac{\dot{J}_1(q_1 a)}{q_1 a J_1(q_1 a)} + \frac{\dot{K}_1(q_2 a)}{q_2 a K_1(q_2 a)} \right)^{-1}$$

$$D = \frac{J_1(q_1 a)}{K_1(q_2 a)} B$$

Luego, escribiendo todas las constantes en respecto a  $B$ :

$$D = \frac{J_1(q_1 a)}{K_1(q_2 a)} B$$

$$A = \frac{\omega\mu_0}{i\beta} \left( \frac{1}{q_2^2 a^2} + \frac{1}{q_1^2 a^2} \right)^{-1} \left( \frac{J_1(q_1 a)}{q_1 a J_1(q_1 a)} + \frac{K_1(q_2 a)}{q_2 a K_1(q_2 a)} \right) B$$

$$C = \frac{J_1(q_1 a)}{K_1(q_2 a)} \frac{\omega\mu_0}{i\beta} \left( \frac{1}{q_2^2 a^2} + \frac{1}{q_1^2 a^2} \right)^{-1} \left( \frac{J_1(q_1 a)}{q_1 a J_1(q_1 a)} + \frac{K_1(q_2 a)}{q_2 a K_1(q_2 a)} \right) B$$

Escribiendo la componente 'z' del campo eléctrico, omitiendo la propagación en 'z' y la variación temporal:

$$E_z = \frac{\omega\mu_0}{i\beta} \left( \frac{1}{q_2^2 a^2} + \frac{1}{q_1^2 a^2} \right)^{-1} \left( \frac{J_1(q_1 a)}{q_1 a J_1(q_1 a)} + \frac{K_1(q_2 a)}{q_2 a K_1(q_2 a)} \right) B J_1(q_1 r) e^{j\theta} \quad r \leq a$$

$$E_z = \frac{J_1(q_1 a)}{K_1(q_2 a)} \frac{\omega\mu_0}{i\beta} \left( \frac{1}{q_2^2 a^2} + \frac{1}{q_1^2 a^2} \right)^{-1} \left( \frac{J_1(q_1 a)}{q_1 a J_1(q_1 a)} + \frac{K_1(q_2 a)}{q_2 a K_1(q_2 a)} \right) B K_1(q_2 r) e^{j\theta} \quad r > a$$

De la misma forma, escribiendo la expresión para el campo magnético:

$$H_z = B J_1(q_1 r) e^{j\theta} \quad r \leq a$$

$$H_z = \frac{J_1(q_1 a)}{K_1(q_2 a)} B K_1(q_2 r) e^{j\theta} \quad r > a$$

Primeramente, graficando la parte real del campo magnético en el núcleo y fijando  $B = 0.859306$  para obtener amplitud unitaria:

```
In [38]: def Hz_HE_11_co(r, theta):
    a = 4.1 * (10 ** -6)
    bb = 0.8593064970502683
    Hz = bb * sp.jv(1, q1(HE_zeros_11)*r) * np.exp((1j)*theta)
```

```
Hzi = np.real(Hz)
return Hzi
```

```
In [39]: #core
dr_HE11, dtheta_HE11 = a/500, (2*np.pi)/500
r_HE11, theta_HE11 = np.mgrid[-a:a+dr_HE11:dr_HE11, 0:2*np.pi+dtheta_HE11:dtheta_HE11]

zz_HE11 = Hz_HE_11_co(r_HE11,theta_HE11)
zz_HE11 = zz_HE11[::-1, ::-1]
z_min_HE11, z_max_HE11 = zz_HE11.min(), zz_HE11.max()
```

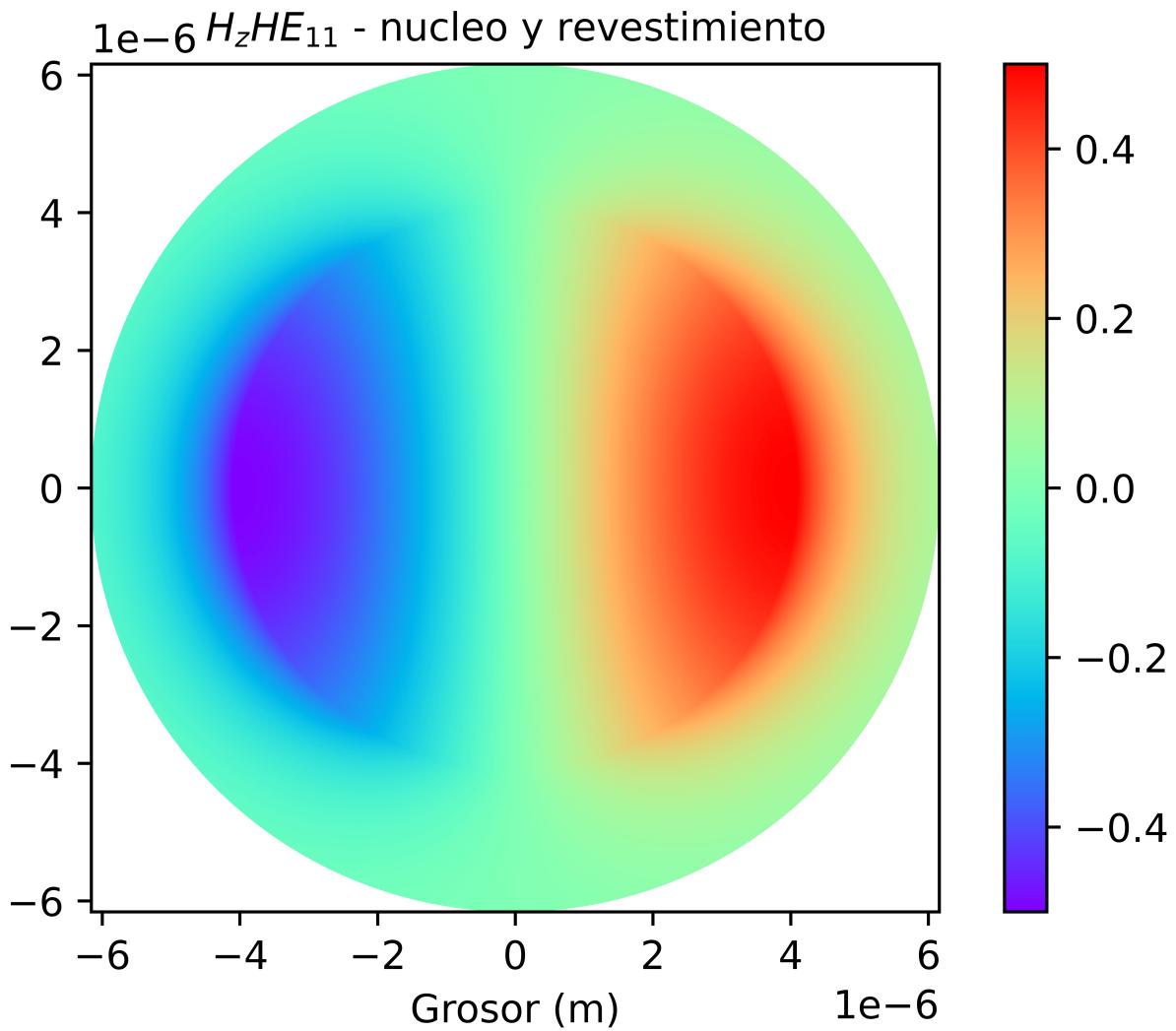
```
In [40]: def Hz_HE_11_cl(r,theta):
    a = 4.1 * (10 ** -6)
    bb = 0.8593064970502683
    Hz = bb * ((sp.jv(1,q1(HE_zeros_11)*a)) / ((sp.kn(1,q2(HE_zeros_11)*a)))) + 1
    Hzi = np.real(Hz)
    return Hzi
```

```
In [41]: #cladding
dr_cl_HE11, dtheta_cl_HE11 = (1*a)/500, (2*np.pi)/500
r_cl_HE11, theta_cl_HE11 = np.mgrid[a:(1.5*a)+dr_cl_HE11:dr_cl_HE11, 0:2*np.pi+dtheta_cl_HE11:dtheta_cl_HE11]

zz_cl_HE11 = Hz_HE_11_cl(r_cl_HE11,theta_cl_HE11)
zz_cl_HE11 = zz_cl_HE11[::-1, ::-1]
z_min_cl_HE11, z_max_cl_HE11 = zz_cl_HE11.min(), zz_cl_HE11.max()
```

```
In [42]: fig, axs = plt.subplots(dpi=1000)
ax = axs
axs.set_aspect('equal', adjustable='box')
c1 = ax.pcolor(polar_car(r_HE11,theta_HE11)[0], polar_car(r_HE11,theta_HE11)[1])
c2 = ax.pcolor(polar_car(r_cl_HE11,theta_cl_HE11)[0], polar_car(r_cl_HE11,theta_cl_HE11)[1])

ax.set(xlabel=r'Grosor (m)',
       title=r'$H_z$ HE_{11} - nucleo y revestimiento')
ax.title.set_size(10)
fig.colorbar(c1, ax=ax)
plt.savefig('Hz_HE11_core_cladding.png')
```



Para la parte electrica del campo en 'z' se tiene que:

$$E_z = \frac{\omega\mu_0}{i\beta} \left( \frac{1}{q_2^2 a^2} + \frac{1}{q_1^2 a^2} \right)^{-1} \left( \frac{J_1(q_1 a)}{q_1 a J_1(q_1 a)} + \frac{K_1(q_2 a)}{q_2 a K_1(q_2 a)} \right) B J_1(q_1 r) e^{j\theta} \quad r \leq a$$

$$E_z = \frac{J_1(q_1 a)}{K_1(q_2 a)} \frac{\omega\mu_0}{i\beta} \left( \frac{1}{q_2^2 a^2} + \frac{1}{q_1^2 a^2} \right)^{-1} \left( \frac{J_1(q_1 a)}{q_1 a J_1(q_1 a)} + \frac{K_1(q_2 a)}{q_2 a K_1(q_2 a)} \right) B K_1(q_2 r) e^{j\theta} \quad r > a$$

Graficando la parte real del campo electrico en el nucleo y teniendo en cuenta que  $B = 0.859306$ :

```
In [43]: def Ez_HE_11_co(r,theta):
    w = (2*np.pi* (3 * (10 ** 6)))
    mu = (4*np.pi* (10 ** -7))
    a = 4.1 * (10 ** -6)
```

```

b = HE_zeros_11
p = (w*mu)/((1j)*(b*(10**6)))
s = ((1/((q2(b)**2)*(a**2))) + (1/((q1(b)**2)*(a**2))))**(-1)
t1 = sp.jvp(1,q1(b)*a,1) / (sp.jv(1,q1(b)*a)*a*q1(b))
t2 = sp.kvp(1,q2(b)*a,1) / (sp.kn(1,q2(b)*a)*q2(b)*a)
t = t1 + t2
bb = 0.8593064970502683
Hz = bb * np.exp((1j)*theta) * sp.jv(1,q1(b)*r) * p * s * t
Hzi = np.real(Hz)
return Hzi

```

In [44]:

```

#core
dr_HE11_Ez, dtheta_HE11_Ez = a/500, (2*np.pi)/500
r_HE11_Ez, theta_HE11_Ez = np.mgrid[-a:a+dr_HE11_Ez:dr_HE11_Ez, 0:2*np.pi+dtheta_HE11_Ez]
zz_HE11_Ez = Ez_HE_11_cl(r_HE11_Ez,theta_HE11_Ez)
zz_HE11_Ez = zz_HE11_Ez[:-1, :-1]
z_min_HE11_Ez, z_max_HE11_Ez = zz_HE11_Ez.min(), zz_HE11_Ez.max()

```

In [45]:

```

def Ez_HE_11_cl(r,theta):
    w = (2*np.pi*(3*(10**6)))
    mu = (4*np.pi*(10**-7))
    a = 4.1*(10**-6)
    b = HE_zeros_11
    p = (w*mu)/((1j)*b)
    s = ((1/((q2(b)**2)*(a**2))) + (1/((q1(b)**2)*(a**2))))**(-1)
    t1 = sp.jvp(1,q1(b)*a,1) / (sp.jv(1,q1(b)*a)*a*q1(b))
    t2 = sp.kvp(1,q2(b)*a,1) / (sp.kn(1,q2(b)*a)*q2(b)*a)
    t = t1 + t2
    f = sp.jv(1,q1(b)*a) / sp.kn(1,q2(b)*a)
    bb = 0.8593064970502683
    Hz = bb * np.exp((1j)*theta) * sp.kn(1,q2(b)*r) * p * s * t * f
    Hzi = np.real(Hz)
    return Hzi

```

In [46]:

```

#cladding
dr_cl_HE11_Ez, dtheta_cl_HE11_Ez = (1*a)/500, (2*np.pi)/500
r_cl_HE11_Ez, theta_cl_HE11_Ez = np.mgrid[a:(1.5*a)+dr_cl_HE11_Ez:dr_cl_HE11_Ez, 0:2*np.pi+dtheta_HE11_Ez]
zz_cl_HE11_Ez = Ez_HE_11_cl(r_cl_HE11_Ez,theta_cl_HE11_Ez)
zz_cl_HE11_Ez = zz_cl_HE11_Ez[:-1, :-1]
z_min_cl_HE11_Ez, z_max_cl_HE11_Ez = zz_cl_HE11_Ez.min(), zz_cl_HE11_Ez.max()

```

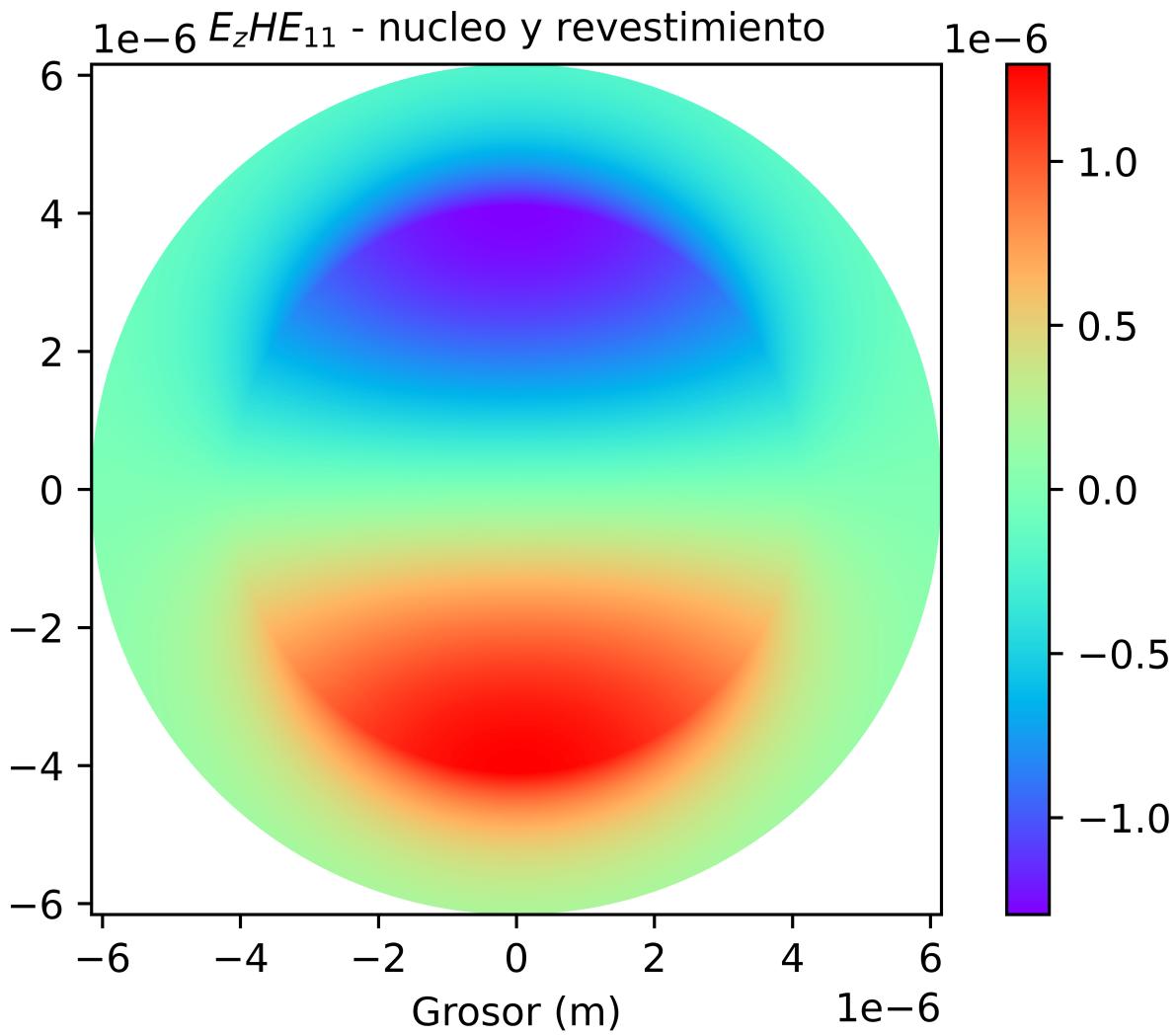
In [47]:

```

fig, axs = plt.subplots(dpi=1000)
ax = axs
ax.set_aspect('equal', adjustable='box')
c1 = ax.pcolor(polar_car(r_HE11_Ez,theta_HE11_Ez)[0], polar_car(r_HE11_Ez,theta_HE11_Ez)[1], zz_HE11_Ez)
c2 = ax.pcolor(polar_car(r_cl_HE11_Ez,theta_cl_HE11_Ez)[0], polar_car(r_cl_HE11_Ez,theta_cl_HE11_Ez)[1], zz_cl_HE11_Ez)

ax.set(xlabel=r'Grosor (m)', title=r'$E_z$ HE_{11} - nucleo y revestimiento')
ax.title.set_size(10)
fig.colorbar(c1, ax=ax)
plt.savefig('Ez_HE11_core_cladding.png')

```



Luego, hallando la norma cuadrada del campo electrico:

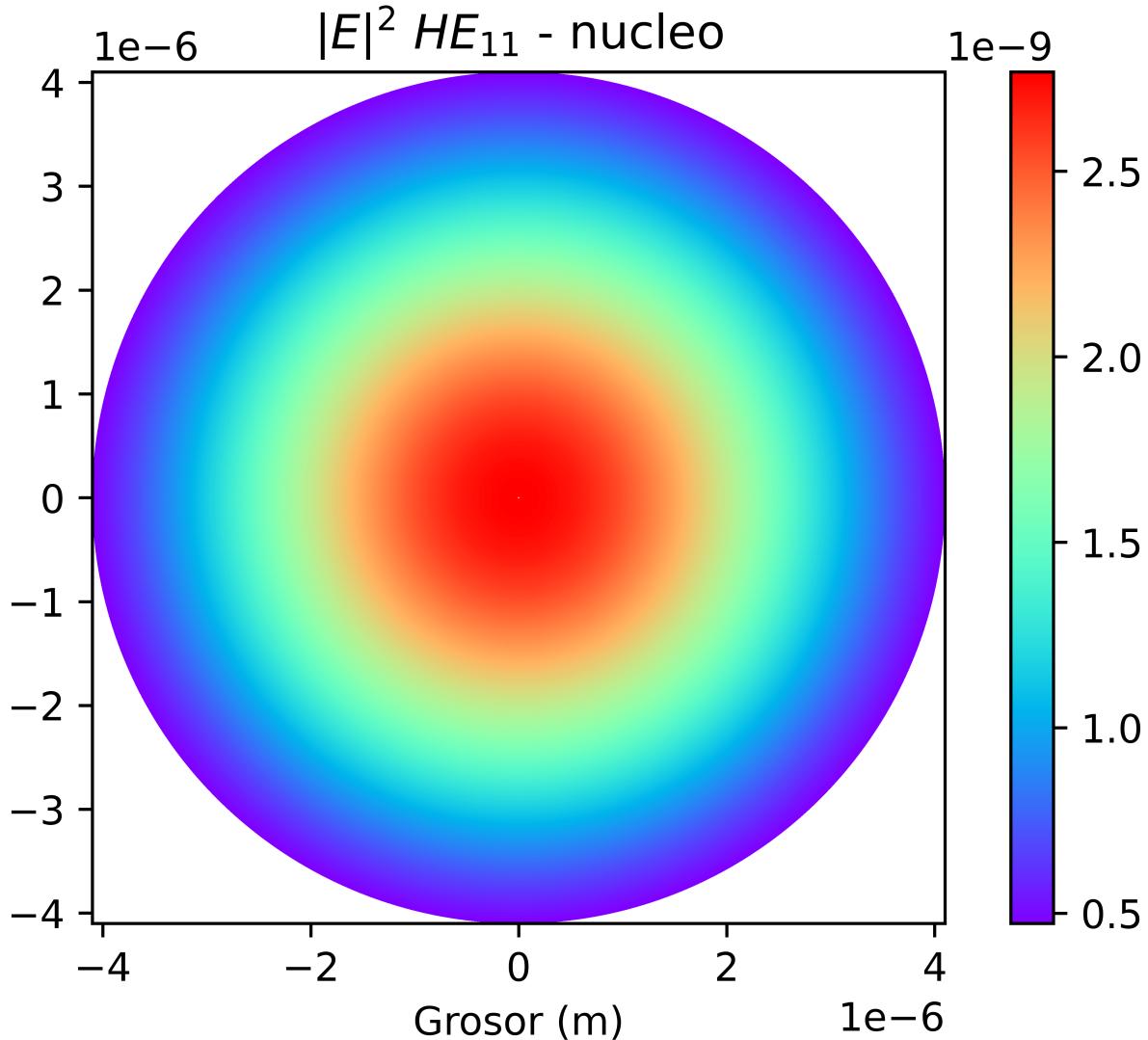
```
In [48]: def normE_co_HE11(r,theta):
    w = (2*np.pi* (3 * (10 ** 6)))
    mu = (4*np.pi* (10 ** -7))
    a = 4.1 * (10 ** -6)
    b = HE_zeros_11
    p = (w*mu)/((1j)*(b * (10**6)))
    s = ((1/((q2(b) ** 2) * (a ** 2))) + (1/((q1(b) ** 2) * (a ** 2)))) ** (-1)
    t1 = sp.jvp(1,q1(b)*a,1) / (sp.jv(1,q1(b)*a) * a * q1(b))
    t2 = sp.kvp(1,q2(b)*a, 1) / (sp.kn(1,q2(b)*a) * q2(b) * a)
    t = t1 + t2
    bb = 0.8593064970502683
    Ez = bb * np.exp((1j)*theta) * sp.jv(1,q1(b)*r) * p * s * t
    Er = (((1j)*(10**6)) / q1(b)) * p * s * t * sp.jvp(1,q1(b)*r,1) * np.exp((1j)*theta)
    Et = ((1j)/(q1(b) ** 2)) * ((-mu*w*q1(b))*sp.jvp(1,q1(b)*r,1)*bb*np.exp((1j)*theta))
    normE2 = Er*np.conj(Er) + Et*np.conj(Et)
    normEz = Ez*np.conj(Ez)
    normEr = Er*np.conj(Er)
    normEt = Et*np.conj(Et)
    return normE2
```

```
In [49]: dr_normE_HE11, dtheta_normE_HE11 = a/500, (2*np.pi)/500
r_normE_HE11, theta_normE_HE11 = np.mgrid[a/500:a+dr_normE_HE11:dr_normE_HE11,
```

```
zz_normE_HE11 = normE_co_HE11(r_normE_HE11,theta_normE_HE11)
zz_normE_HE11 = np.real(zz_normE_HE11[:-1, :-1])
z_min_normE_HE11, z_max_normE_HE11 = zz_normE_HE11.min(), zz_normE_HE11.max()
```

```
In [50]: fig, axs = plt.subplots(dpi=1000)
ax = axs
axs.set_aspect('equal', adjustable='box')
c1 = ax.pcolor(polar_car(r_normE_HE11,theta_normE_HE11)[0], polar_car(r_normE_HE11,theta_normE_HE11)[1], zz_normE_HE11, cmap='jet')
ax.set(xlabel=r'Grosor (m)', title=r'$|E|^2 HE_{11} - nucleo$')
fig.colorbar(c1, ax=ax)
plt.savefig('NormE_HE11_core.png')
```



## Modo $HE_{21}$

Las relaciones entre las constantes del modo híbrido  $HE_{21}$  quedarían de la siguiente manera:

$$D = \frac{J_2(q_1 a)}{K_2(q_2 a)} B$$

$$A = \frac{\omega\mu_0}{2i\beta} \left( \frac{1}{q_2^2 a^2} + \frac{1}{q_1^2 a^2} \right)^{-1} \left( \frac{J_2'(q_1 a)}{q_1 a J_2(q_1 a)} + \frac{K_2'(q_2 a)}{q_2 a K_2(q_2 a)} \right) B$$

$$C = \frac{J_2(q_1 a)}{K_2(q_2 a)} \frac{\omega\mu_0}{2i\beta} \left( \frac{1}{q_2^2 a^2} + \frac{1}{q_1^2 a^2} \right)^{-1} \left( \frac{J_2'(q_1 a)}{q_1 a J_2(q_1 a)} + \frac{K_2'(q_2 a)}{q_2 a K_2(q_2 a)} \right) B$$

Escribiendo la expresión para el campo magnético:

$$H_z = B J_2(q_1 r) e^{2j\theta} \quad r \leq a$$

$$H_z = \frac{J_2(q_1 a)}{K_2(q_2 a)} B K_2(q_2 r) e^{2j\theta} \quad r > a$$

Primeramente, graficando la parte real del campo magnético en el nucleo y fijando  $B = 1.036228$  para obtener amplitud unitaria:

```
In [51]: def Hz_HE_21_co(r, theta):
    a = 4.1 * (10 ** -6)
    bb= 1.0362280494516762
    b = HE_zeros_21
    Hz = bb * sp.jv(2,q1(b)*r) * np.exp(2 * (1j) *theta)
    Hzi = np.real(Hz)
    return Hzi
```

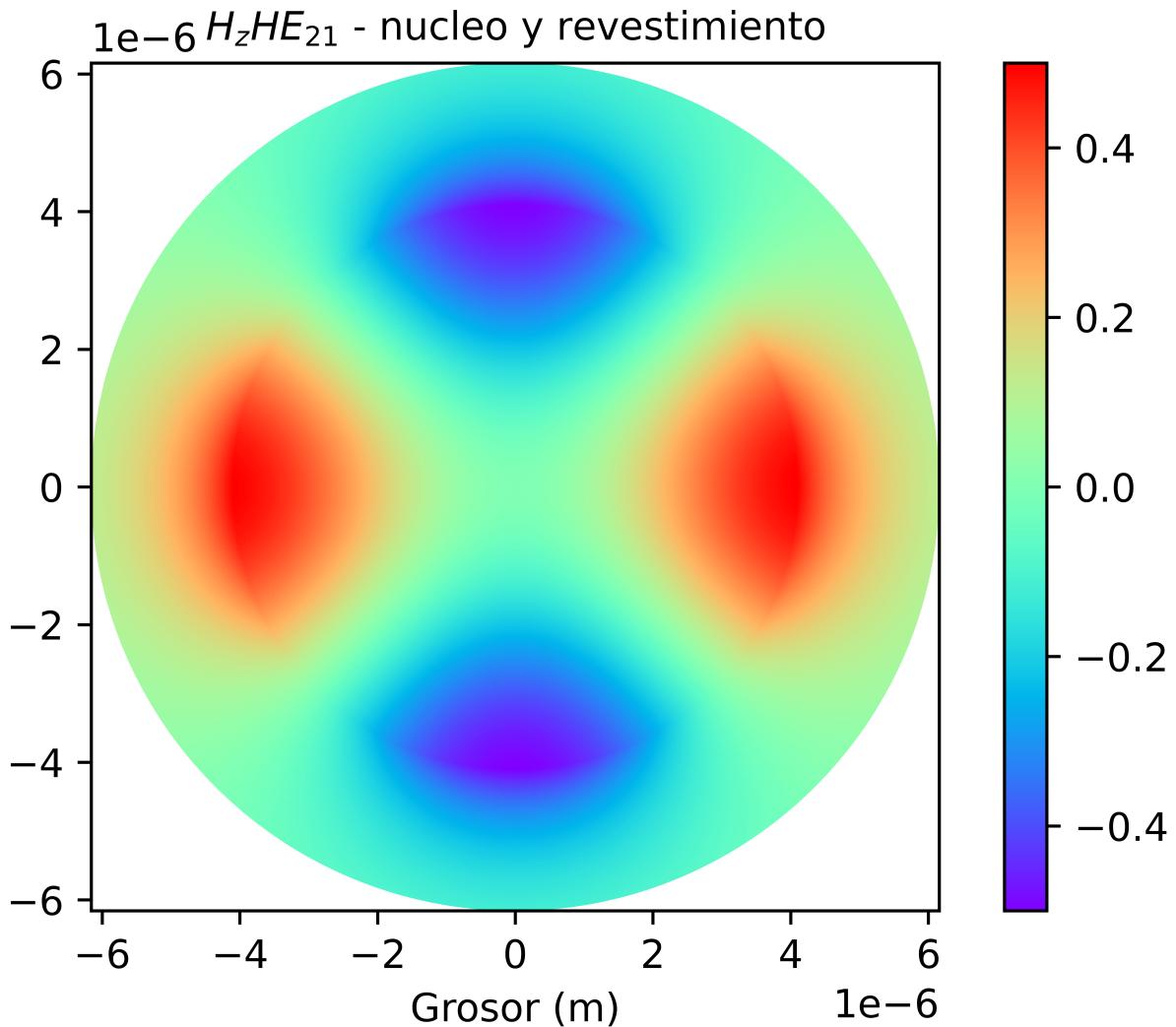
```
In [52]: #core
dr_HE21, dtheta_HE21 = a/500, (2*np.pi)/500
r_HE21, theta_HE21 = np.mgrid[-a:a+dr_HE21:dr_HE21, 0:2*np.pi+dtheta_HE21:dtheta_HE21]
zz_HE21 = Hz_HE_21_co(r_HE21,theta_HE21)
zz_HE21 = zz_HE21[:, :-1, :-1]
z_min_HE21, z_max_HE21 = zz_HE21.min(), zz_HE21.max()
```

```
In [53]: def Hz_HE_21_cl(r, theta):
    a = 4.1 * (10 ** -6)
    bb = 1.0362280494516762
    b = HE_zeros_21
    Hz = bb * ((sp.jv(2,q1(b)*a)) / ((sp.kn(2,q2(b)*a)))) * sp.kn(2,q2(b)*r) *
    Hzi = np.real(Hz)
    return Hzi
```

```
In [54]: #cladding
dr_cl_HE21, dtheta_cl_HE21 = (1*a)/500, (2*np.pi)/500
```

```
r_cl_HE21, theta_cl_HE21 = np.mgrid[a:(1.5*a)+dr_cl_HE21:dr_cl_HE21, 0:2*np.pi]
zz_cl_HE21 = Hz_HE_21_cl(r_cl_HE21,theta_cl_HE21)
zz_cl_HE21 = zz_cl_HE21[:-1, :-1]
z_min_cl_HE21, z_max_cl_HE21 = zz_cl_HE21.min(), zz_cl_HE21.max()
```

```
In [55]: fig, axs = plt.subplots(dpi=1000)
ax = axs
ax.set_aspect('equal', adjustable='box')
c1 = ax.pcolor(polar_car(r_HE21,theta_HE21)[0], polar_car(r_HE21,theta_HE21)[1],
                zz_cl_HE21, cmap='jet')
c2 = ax.pcolor(polar_car(r_cl_HE21,theta_cl_HE21)[0], polar_car(r_cl_HE21,theta_cl_HE21)[1],
                zz_cl_HE21, cmap='jet')
ax.set(xlabel=r'Grosor (m)',
       title=r'$H_z$ HE21 - nucleo y revestimiento')
ax.title.set_size(10)
fig.colorbar(c1, ax=ax)
plt.savefig('Hz_HE21_core_cladding.png')
```



Para la parte electrica del campo en 'z' se tiene que:

$$E_z = \frac{\omega\mu_0}{2i\beta} \left( \frac{1}{q_2^2 a^2} + \frac{1}{q_1^2 a^2} \right)^{-1} \left( \frac{J_2(q_1 a)}{q_1 a J_2(q_1 a)} + \frac{K_2(q_2 a)}{q_2 a K_2(q_2 a)} \right) B J_2(q_1 r) e^{2j\theta} \quad r \leq a$$

$$E_z = \frac{J_2(q_1 a)}{K_2(q_2 a)} \frac{\omega \mu_0}{2i\beta} \left( \frac{1}{q_2^2 a^2} + \frac{1}{q_1^2 a^2} \right)^{-1} \left( \frac{J_2(q_1 a)}{q_1 a J_2(q_1 a)} + \frac{K_2(q_2 a)}{q_2 a K_2(q_2 a)} \right) B K_2(q_2 r) e^{2j\theta} \quad r > a$$

Graficando la parte real del campo electrico en el nucleo y teniendo en cuenta que

$B = 1.036228$ :

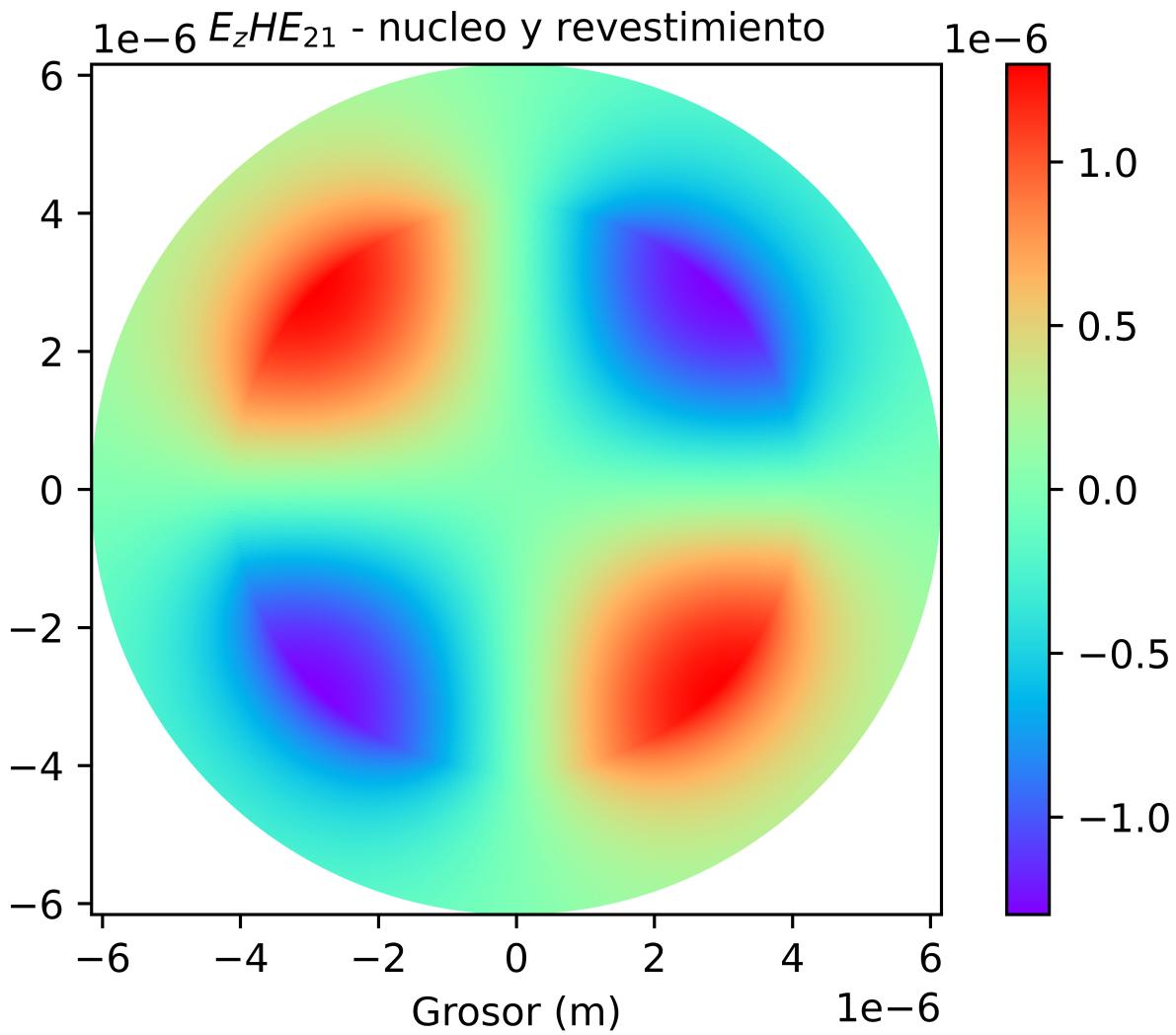
```
In [56]: def Ez_HE_21_co(r, theta):
    w = (2*np.pi* (3 * (10 ** 6)))
    mu = (4*np.pi* (10 ** -7))
    a = 4.1 * (10 ** -6)
    b = HE_zeros_21
    p = (w*mu)/(2 * (1j) *(b*(10**6)))
    s = ((1/((q2(b) ** 2) * (a ** 2))) + (1/((q1(b) ** 2) * (a ** 2)))) ** (-1)
    t1 = sp.jvp(2,q1(b)*a,1) / (sp.jv(2,q1(b)*a) * a * q1(b))
    t2 = sp.kvp(2,q2(b)*a, 1) / (sp.kn(2,q2(b)*a) * q2(b) * a)
    t = t1 + t2
    bb = 1.0362280494516762
    Hz = bb * np.exp(2 * (1j) * theta) * sp.jv(2,q1(b)*r) * p * s * t
    Hzi = np.real(Hz)
    return Hzi
```

```
In [57]: #core
dr_HE21_Ez, dtheta_HE21_Ez = a/500, (2*np.pi)/500
r_HE21_Ez, theta_HE21_Ez = np.mgrid[-a:a+dr_HE21_Ez:dr_HE21_Ez, 0:2*np.pi+dtheta_HE21_Ez]
zz_HE21_Ez = Ez_HE_21_co(r_HE21_Ez,theta_HE21_Ez)
zz_HE21_Ez = zz_HE21_Ez[:-1, :-1]
z_min_HE21_Ez, z_max_HE21_Ez = zz_HE21_Ez.min(), zz_HE21_Ez.max()
#0.5/4.697029354881661, z_max_HE11_Ez*0.10645026083993832
```

```
In [58]: def Ez_HE_21_cl(r, theta):
    w = (2*np.pi* (3 * (10 ** 6)))
    mu = (4*np.pi* (10 ** -7))
    a = 4.1 * (10 ** -6)
    b = HE_zeros_21
    p = (w*mu)/(2 * (1j) *(b*(10**6)))
    s = ((1/((q2(b) ** 2) * (a ** 2))) + (1/((q1(b) ** 2) * (a ** 2)))) ** (-1)
    t1 = sp.jvp(2,q1(b)*a,1) / (sp.jv(2,q1(b)*a) * a * q1(b))
    t2 = sp.kvp(2,q2(b)*a, 1) / (sp.kn(2,q2(b)*a) * q2(b) * a)
    t = t1 + t2
    f = sp.jv(2,q1(b)*a) / sp.kn(2,q2(b)*a)
    bb = 1.0362280494516762
    Hz = bb * np.exp(2 * (1j) *theta) * sp.kn(2,q2(b)*r) * p * s * t * f
    Hzi = np.real(Hz)
    return Hzi
```

```
In [59]: #cladding
dr_cl_HE21_Ez, dtheta_cl_HE21_Ez = (1*a)/500, (2*np.pi)/500
r_cl_HE21_Ez, theta_cl_HE21_Ez = np.mgrid[a:(1.5*a)+dr_cl_HE21_Ez:dr_cl_HE21_Ez]
zz_cl_HE21_Ez = Ez_HE_21_cl(r_cl_HE21_Ez,theta_cl_HE21_Ez)
zz_cl_HE21_Ez = zz_cl_HE21_Ez[:-1, :-1]
z_min_cl_HE21_Ez, z_max_cl_HE21_Ez = zz_cl_HE21_Ez.min(), zz_cl_HE21_Ez.max()
```

```
In [60]: fig, axs = plt.subplots(dpi=1000)
ax = axs
axs.set_aspect('equal', adjustable='box')
c1 = ax.pcolor(polar_car(r_HE21_Ez,theta_HE21_Ez)[0], polar_car(r_HE21_Ez,theta_HE21_Ez)[1], polar_car(r_cl_HE21_Ez,theta_cl_HE21_Ez)[0], polar_car(r_cl_HE21_Ez,theta_cl_HE21_Ez)[1])
ax.set(xlabel=r'Grosor (m)', title=r'$E_z$ HE21 - nucleo y revestimiento')
ax.title.set_size(10)
fig.colorbar(c1, ax=ax)
plt.savefig('Ez_HE21_core_cladding.png')
```



```
In [61]: def normE_co_HE21(r,theta):
    w = (2*np.pi* (3 * (10 ** 6)))
    mu = (4*np.pi* (10 ** -7))
    a = 4.1 * (10 ** -6)
    b = HE_zeros_21
    p = (w*mu)/(2 * (1j) *(b*(10**6)))
    s = ((1/((q2(b) ** 2) * (a ** 2))) + (1/((q1(b) ** 2) * (a ** 2)))) ** (-1)
    t1 = sp.jvp(2,q1(b)*a,1) / (sp.jv(2,q1(b)*a) * a * q1(b))
    t2 = sp.kvp(2,q2(b)*a, 1) / (sp.kn(2,q2(b)*a) * q2(b) * a)
    t = t1 + t2
    bb = 1.0362280494516762
    Ez = bb * np.exp(2 * (1j) * theta) * sp.jv(2,q1(b)*r) * p * s * t
    Er = (((1j)*(b*(10**6))) / q1(b)) * p * s * t * sp.jvp(2,q1(b)*r,1) * np.e
```

```
Et = ((1j)/(q1(b) ** 2)) * (((-mu*w*q1(b)*sp.jvp(2,q1(b)*r,1)*bb*np.exp(2*(1j)*r)/normE2))
normE2 = np.sqrt(Ez*np.conj(Ez) + Er*np.conj(Er) + Et*np.conj(Et))
return normE2
```

In [62]:

```
dr_normE_HE21, dtheta_normE_HE21 = a/500, (2*np.pi)/500
r_normE_HE21, theta_normE_HE21 = np.mgrid[a/500:a+dr_normE_HE21:dr_normE_HE21,
                                             0:dtheta_normE_HE21: dtheta_normE_HE21]
zz_normE_HE21 = normE_co_HE21(r_normE_HE21, theta_normE_HE21)
zz_normE_HE21 = np.real(zz_normE_HE21[:, :-1, :-1])
z_min_normE_HE21, z_max_normE_HE21 = zz_normE_HE21.min(), zz_normE_HE21.max()
```

In [63]:

```
fig, axs = plt.subplots(dpi=1000)
ax = axs
axs.set_aspect('equal', adjustable='box')
c1 = ax.pcolor(polar_car(r_normE_HE21,theta_normE_HE21)[0], polar_car(r_normE_HE21,theta_normE_HE21)[1],
                zz_normE_HE21, cmap='jet')
ax.set(xlabel=r'Grosor (m)', title=r'$|E|^2 HE_{21} - nucleo$')
fig.colorbar(c1, ax=ax)
plt.savefig('NormE_HE21_core.png')
```

