0.1 Utilización mallas Tesis Milan

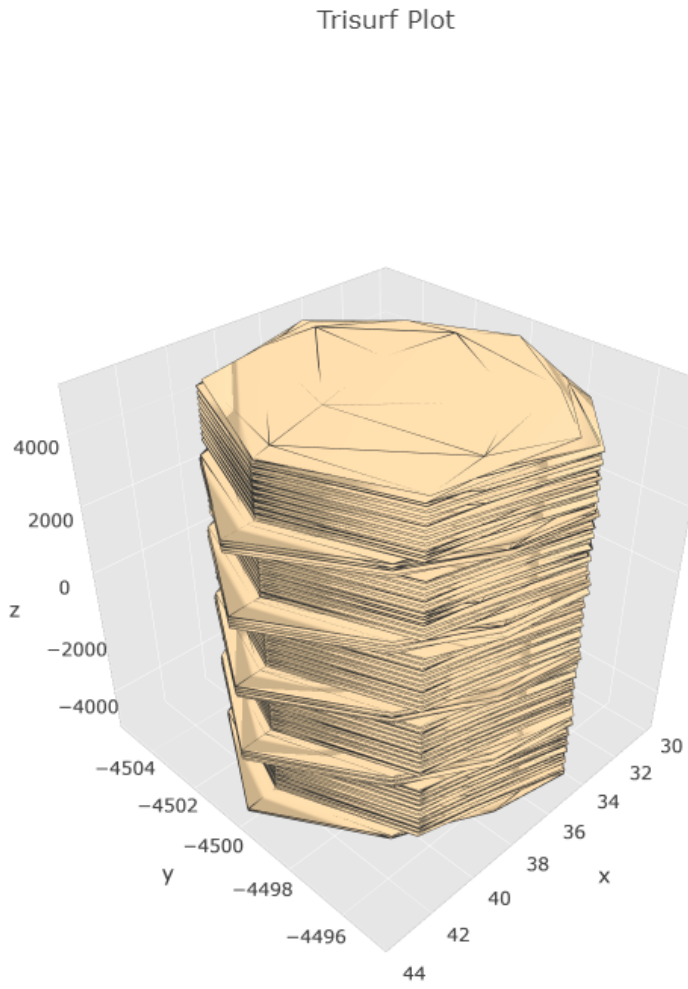

Figura 0.1: Vista malla en Jupyter Notebook

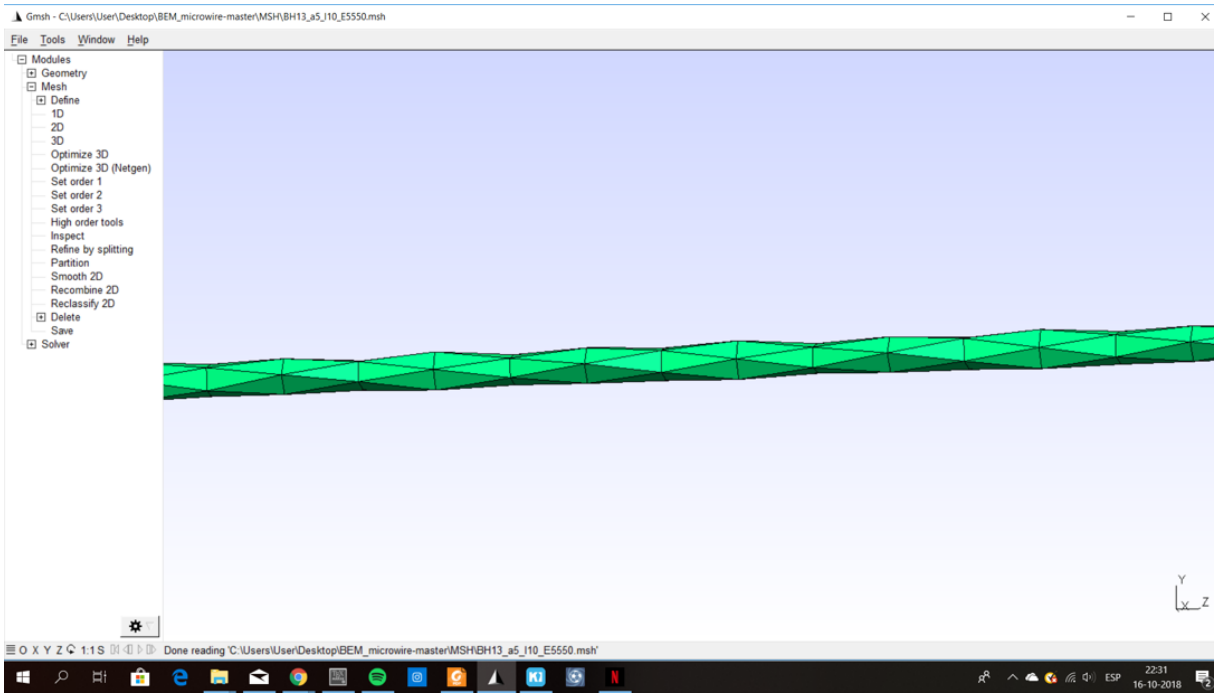

Figura 0.2: Vista malla en GSHM (sección)

Resultados:

0.1.1 Ecuación 1.

$$\begin{bmatrix} \frac{1}{2}Id + S & -D \\ \frac{1}{2}Id - S & \frac{\mu_2}{\mu_1}D \end{bmatrix} \begin{bmatrix} u_1 \\ \frac{\partial u_1}{\partial n} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\mu_1 - \mu_2}{\mu_1} \frac{\partial u_{inc}}{\partial n} \end{bmatrix}$$

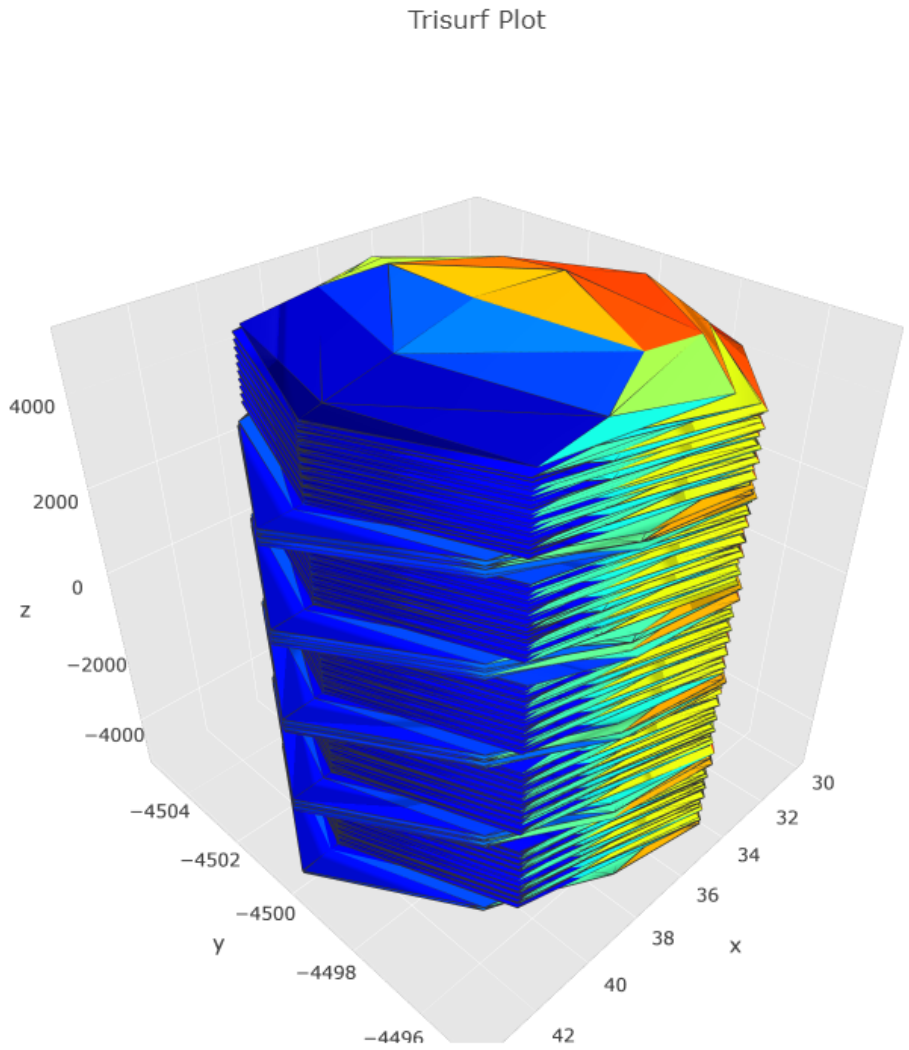

Figura 0.3: Solución graficada de la ecuación 1

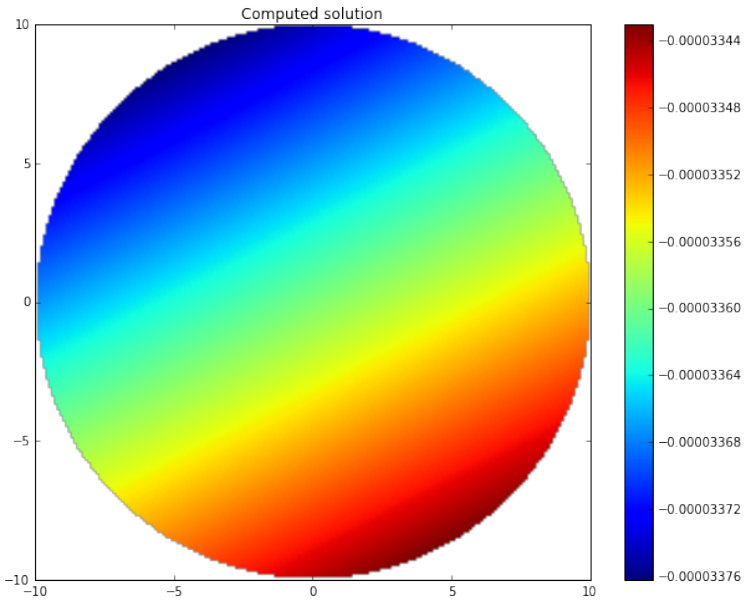

Figura 0.4: 'Corte' de la solución en $z = 0$

0.1.2 Ecuación 2.

$$\begin{bmatrix} -D_{ext} - D_{int} & \alpha S_{int} + S_{ext} \\ -D'_{ext} - D'_{int} & \frac{\alpha - 1}{2} + \alpha S'_{int} + S'_{ext} \end{bmatrix} \begin{bmatrix} u^{ext} \\ \frac{\partial u^{ext}}{\partial n} \end{bmatrix} = \begin{bmatrix} u_{inc} \\ \frac{\partial u_{inc}}{\partial n} \end{bmatrix}$$

```
In [31]: #Problem data
         omega = 2.*np.pi*10.e9
         e0 = 8.854*1e-12*1e-18
         mu0 = 4.*np.pi*1e-7*1e6
         mue = (1.)*mu0
         ee = (16.)*e0
         mui = (-2.9214+0.5895j)*mu0
         ei = (82629.2677-200138.2211j)*e0
         k = omega*np.sqrt(e0*mu0)
         lam = 2*np.pi/k
         nm = np.sqrt((ee*mue)/(e0*mu0))
         nc = np.sqrt((ei*mui)/(e0*mu0))
         alfa_m = mue/mu0
         alfa_c = mui/mue
         antena = np.array([[1e4],[0.],[0.]])
         print("Numero de onda exterior:", k)
         print("Indice de refraccion matriz:", nm)
         print("Indice de refraccion conductor:", nc)
         print("Numero de onda interior matriz:", nm*k)
         print("Numero de onda interior conductor:", nm*nc*k)
         print("Indice de transmision matriz:", alfa_m)
         print("Indice de transmision conductor:", alfa_c)
         print("Longitud de onda:", lam, "micras")
```

```
Numero de onda exterior: 0.0002095822793
Indice de refraccion matriz: 4.0
Indice de refraccion conductor: (510.829219424+619.966251289j)
Numero de onda interior matriz: 0.000838329117198
Numero de onda interior conductor: (0.428243008559+0.519735760136j)
Indice de transmision matriz: 1.0
Indice de transmision conductor: (-2.9214+0.5895j)
Longitud de onda: 29979.5637693 micras
```

```
In [32]: #Dirichlet and Neumann functions
         def dirichlet_fun(x, n, domain_index, result):
                 result[0] = 1. * np.exp(1j * k * x[1])
         def neumann_fun(x, n, domain_index, result):
                 result[0] = 1. * 1j * k * n[1] * np.exp(1j * k * x[1])
```

```
In [33]: #Multitrace Operators
Ai_0 = bempp.api.operators.boundary.helmholtz.multitrace_operator(grid_0, nm * nc * k)
Ae_0 = bempp.api.operators.boundary.helmholtz.multitrace_operator(grid_0, nm * k)

         #Multitrace Transmission
         Ai_0[0,1] = Ai_0[0,1]*alfa_c
         Ai_0[1,1] = Ai_0[1,1]*alfa_c

         #Interior and exterior link up
         op_0 = (Ai_0 + Ae_0)
```

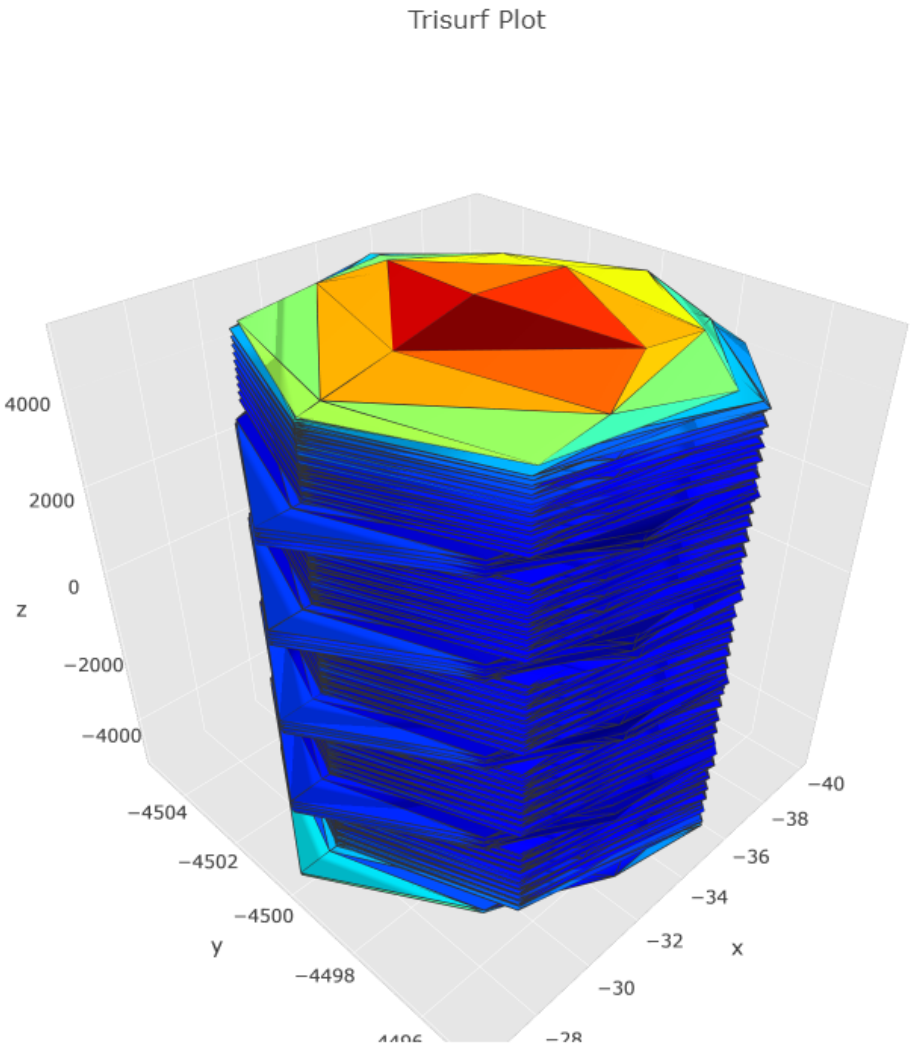

Figura 0.5: Solución graficada de la ecuación 2

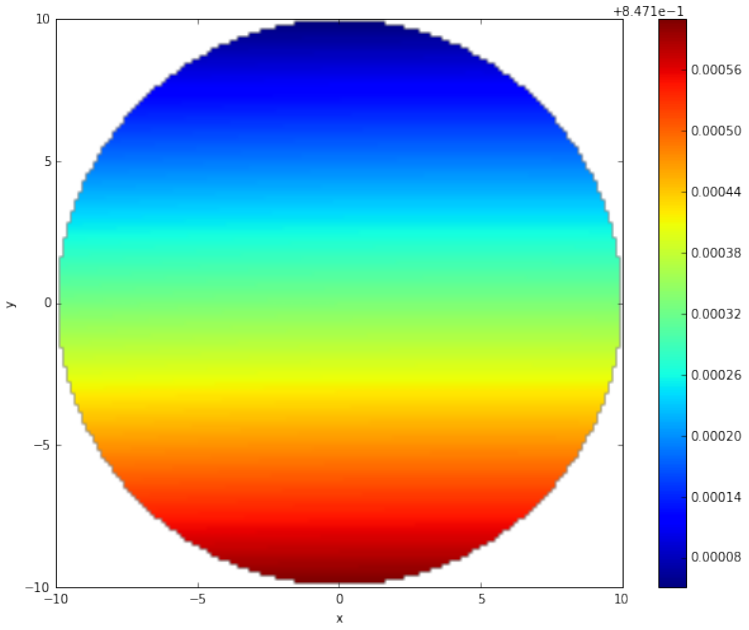

Figura 0.6: 'Corte' de la solución en $z = 0$