

# Diagramme de rayonnement

Ahmed Ammar (ahmed.ammar@fst.utm.tn)

Institut Préparatoire aux Études Scientifiques et Techniques, Université de Carthage.

Feb 24, 2020

## 1 Antenne dipolaire

Pour une antenne dipolaire de longueur  $L$ , le calcul est très complexe car on ne connaît pas a priori l'expression de l'intensité  $I(z,t)$  du courant dans l'antenne. Les calculs conduisent à l'expression suivante :

$$I(z,t) = I_0 \sin\left(\frac{L}{2} - z\right) e^{-i\omega t} \quad (1)$$

On par ailleurs  $I(-z,t)=I(z,t)$ . Le courant s'annule aux extrémités et varie sinusoidalement avec une période  $\lambda$ . Connaissant le courant, on peut calculer le champ électromagnétique en sommant les contributions des segments élémentaires, qui sont des dipôles oscillants.

Pour une antenne de longueur petite devant la longueur d'onde, le courant décroît linéairement entre sa valeur  $I_0$  au centre de l'antenne et une valeur nulle à l'extrémité. Dans ce cas, on peut utiliser les résultats du dipôle oscillant en remplaçant  $I_0$  par  $I_0/2$ .

Il est intéressant d'augmenter la longueur des antennes car la puissance émise est proportionnelle au carré de la longueur. Pour des fréquences supérieures à  $100 \text{ MHz}$ , on utilise des antennes dont la longueur n'est pas petite devant la longueur d'onde. Par exemple, une antenne demi-onde a une longueur égale à  $\lambda/2$ . Dans le cas général, le facteur angulaire du champ électrique est la fonction suivante :

$$f(\theta) = \frac{\cos\left(\frac{kL}{2} \cos\theta\right) - \cos\left(\frac{kL}{2}\right)}{\sin\theta} \quad (2)$$

Le carré de cette fonction permet de tracer le diagramme de rayonnement en fonction du rapport

$$\frac{kL}{2} = \frac{\pi L}{\lambda} \quad (3)$$

### 1.1 Simulation Python

```

## NOM DU PROGRAMME: DipRadiation.py
%% IMPORTATION
import numpy as np
import matplotlib.pyplot as plt
import mpl_toolkits.mplot3d.axes3d as axes3d

def rho(theta, rapport = 0.5):
    u = rapport* np.pi
    F = (np.cos(u*np.cos(theta)) - np.cos(u))/(np.sin(theta))
    G = F * F
    return G/G.max()

plt.figure()
ax = plt.subplot(111, polar=True)
plt.title("Rayonnement d'une antenne dipolaire de longueur L")
ax.set_theta_zero_location('N')
ax.set_theta_direction(-1)

ax.grid(True)
theta = np.linspace(0.01,2*np.pi,500)

for rapport in [0.5,1.0,1.5, 2]:
    ax.plot(theta,rho(theta, rapport), lw = 2,
            label=r"$L/\lambda=%.1f$"%rapport)
plt.legend(loc='lower right')
plt.tight_layout()
plt.savefig("dipole.png"); plt.savefig("dipole.pdf")
plt.show()

%% Dipôle 3D
Theta = np.linspace(0.001,np.pi,400)
Phi = np.linspace(0.001,2*np.pi,400)
THETA, PHI = np.meshgrid(Theta,Phi)
#
def sph2cart(azimuth,elevation,r):
    """
    Convertisseur de Coordonnée Sphérique/Cartésienne
    """
    x = r * np.sin(elevation) * np.cos(azimuth)
    y = r * np.sin(elevation) * np.sin(azimuth)
    z = r * np.cos(elevation)
    return x, y, z

import matplotlib.colors as mcolors
fig = plt.figure(figsize=(8,5))
cmap = plt.get_cmap('gnuplot')
rapport = 1.4
X, Y, Z = sph2cart(PHI,THETA,rho(THETA,rapport))
#ax1 = plt.subplot(121, polar=True)
#ax1.plot(theta,rho(theta, rapport), lw = 2)
#ax1.set_theta_zero_location('N')
#ax1.set_theta_direction(-1)
ax2 = plt.subplot(111, projection='3d')
ax2._axis3don = False # hide x, y, z axis
norm = mcolors.Normalize(vmin=Z.min(), vmax=Z.max())
ax2.plot_surface(X, Y, Z, rstride=8, cstride=8,
                 facecolors=cmap(norm(Z)), antialiased=True, alpha=0.5)
fig.suptitle("Diagramme de rayonnement: "+r"$L/\lambda=%.1f$"%rapport)
plt.tight_layout()
plt.savefig("dipole3D_w14.png"); plt.savefig("dipole3D_w14.pdf")

```

Rayonnement d'une antenne dipolaire de longueur  $L$

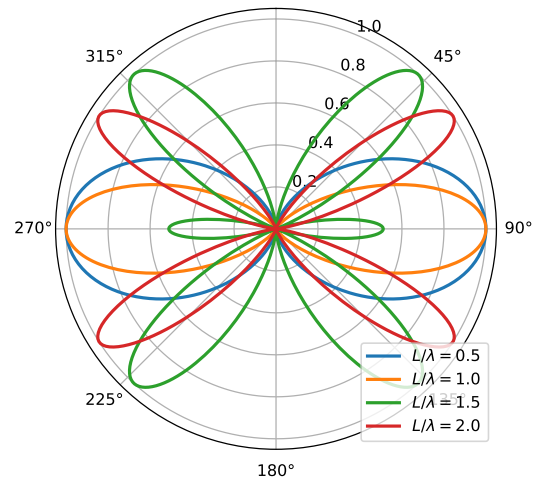


Diagramme de rayonnement:  $L/\lambda = 0.5$

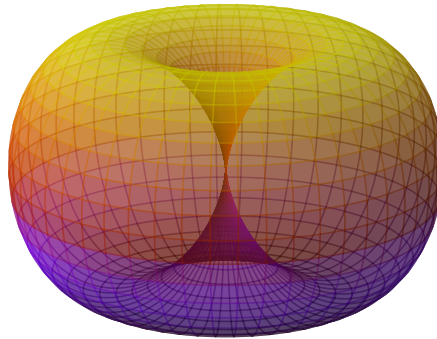


Diagramme de rayonnement:  $L/\lambda = 1.4$

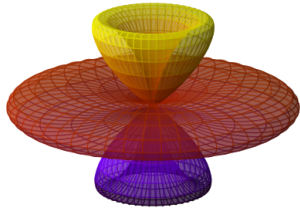


Diagramme de rayonnement:  $L/\lambda = 1.5$

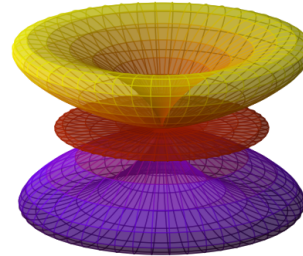


Diagramme de rayonnement:  $L/\lambda = 2.0$

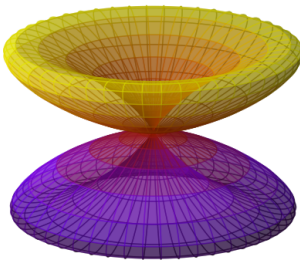


Diagramme de rayonnement:  $L/\lambda = 2.4$

