1 Modelling An Antenna Array: Using Python programming language- Numpy and Matplotlib

Directing radio waves in a particular direction by adjusting their number, geometrical arrangement, and relative amplitudes and phases.

Author: Jay Gautam, Darmstadt Germany

1.1 Two-antenna case (formulae and theory below)

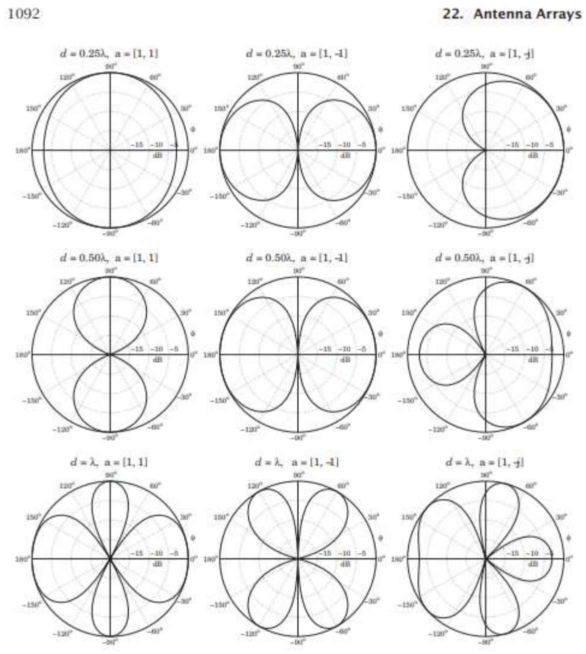
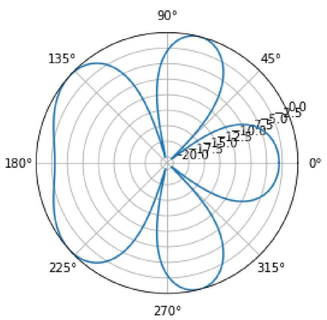


Fig. 22.3.1 Azimuthal gain patterns of two-element isotropic array.

In [13]:

```python
import numpy as np
import matplotlib.pyplot as plt

# Calculation of : Array factor and gain.
def gain(d, w):
    phi = np.linspace(0, 2*np.pi, 1000)
    psi = 2*np.pi * d / lam * np.cos(phi) # psi = (2*pi*d/lamda)*cos(phi)
    A = w[0] + w[1]*np.exp(1j*psi) # Array factor for two antenna case.
    g = np.abs(A)**2 # relative radiation power pattern ("gain") is the square of Array factor.
    return phi, g

# Calculation of the dirrective gain (dbi scale) of the antenna array.
def get_directive_gain(g, minDdBi=-20):
    DdBi = 10 * np.log10(g / np.max(g)) # directive gain = 10*log_10(g/g_max)
    return np.clip(DdBi, minDdBi, None) # It clips the directive gain below the certain values (minDdBi=-20).

# Wavelength(lam), antenna spacing(d), feed coefficients(w). => It determines the shape of the radiation pattern
lam = 1
d = lam
w = np.array([1, -1j]) # w[0] = 1, w[1] = -1j

# gain and directive gain.
phi, g = gain(d, w)
DdBi = get_directive_gain(g)

# Polar plot.
plt.polar(phi, DdBi)
# ax = plt.gca()
# ax.set_rticks([-20, -15, -10, -5])
# ax.set_rlabel_position(45)
plt.show()
```



1.2 Three-antenna array case (formulae and theory below)

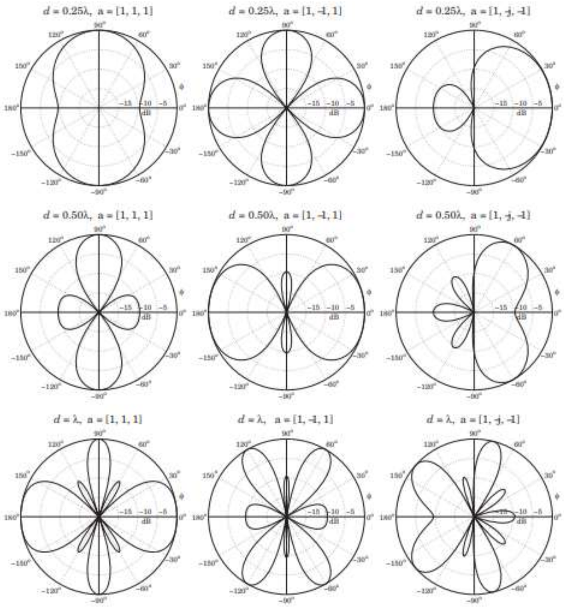
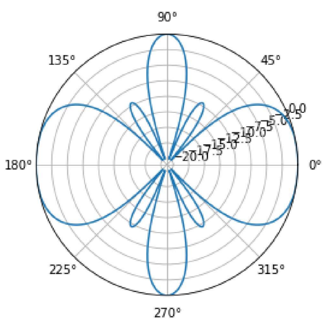


Fig. 22.3.3 Azimuthal gains of three-element isotropic array.

In [14]:

```python
import numpy as np
import matplotlib.pyplot as plt

# Calculation of : Array factor and gain- in case of 3 array elements.
def gain(d, w):
    """Return the power as a function of azimuthal angle, phi."""
    phi = np.linspace(0, 2*np.pi, 1000)
    psi = 2*np.pi * d / lam * np.cos(phi)
    j = np.arange(len(w))
    A = np.sum(w[j] * np.exp(j * 1j * psi[:, None]), axis=1)
    g = np.abs(A)**2
    return phi, g

# Calculation of the dirrective gain (dbi scale) of the 3 antenna array.
def get_directive_gain(g, minDdBi=-20):
    """Return the "directive gain" of the antenna array producing gain g."""
    DdBi = 10 * np.log10(g / np.max(g))
    return np.clip(DdBi, minDdBi, None)

# Wavelength(lam), antenna spacing(d), feed coefficients(w) => It determines the shape of the radiation pattern
lam = 1
#d = lam / 2
d = lam
#w = np.array([1, -1, 1])
w = np.array([1, 1, 1])
#w = np.array([1, 1, 1, 1j])

# gain and directive gain.
phi, g = gain(d, w)
DdBi = get_directive_gain(g)

# Polar plot.
fig = plt.figure()
ax = fig.add_subplot(projection='polar')
ax.plot(phi, DdBi)
# ax.set_rticks([-20, -15, -10, -5])
# ax.set_rlabel_position(45)
plt.show()
```



1.3 Theory and formulae

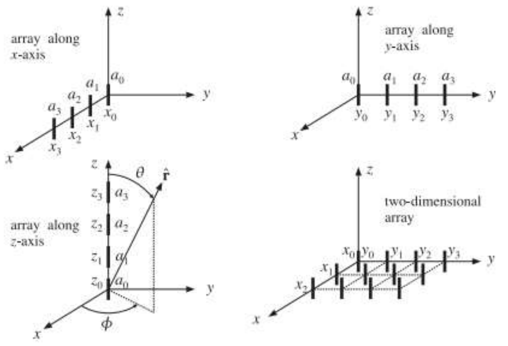


Fig. 22.1.1 Typical array configurations.

More generally, we consider a three-dimensional array of several identical antennas located at positions $\mathbf{d}_0, \mathbf{d}_1, \mathbf{d}_2, \dots$ with relative feed coefficients $a_0, a_1, a_2, \dots$, as shown in Fig. 22.2.1. (Without loss of generality, we may set $\mathbf{d}_0 = 0$ and $a_0 = 1$.)

The current density of the nth antenna will be $\boldsymbol{J}_n(\boldsymbol{r}) = a_n \boldsymbol{J}(\boldsymbol{r} - \mathbf{d}_n)$ and the corresponding radiation vector:

$$\boldsymbol{F}_n(\mathbf{k}) = a_n e^{j\mathbf{k}\cdot\mathbf{d}_n} \boldsymbol{F}(\mathbf{k})$$

The total current density of the array will be:

$$\boldsymbol{J}_{\text{tot}}(\boldsymbol{r}) = a_0 \boldsymbol{J}(\boldsymbol{r} - \mathbf{d}_0) + a_1 \boldsymbol{J}(\boldsymbol{r} - \mathbf{d}_1) + a_2 \boldsymbol{J}(\boldsymbol{r} - \mathbf{d}_2) + \cdots$$

and the total radiation vector:

$$\boldsymbol{F}_{\text{tot}}(\mathbf{k}) = \boldsymbol{F}_0 + \boldsymbol{F}_1 + \boldsymbol{F}_2 + \cdots = a_0 e^{j\mathbf{k}\cdot\mathbf{d}_0} \boldsymbol{F}(\mathbf{k}) + a_1 e^{j\mathbf{k}\cdot\mathbf{d}_1} \boldsymbol{F}(\mathbf{k}) + a_2 e^{j\mathbf{k}\cdot\mathbf{d}_2} \boldsymbol{F}(\mathbf{k}) + \cdots$$

The factor $\boldsymbol{F}(\mathbf{k})$ due to a single antenna element at the origin is common to all terms. Thus, we obtain the *array pattern multiplication* property:

$$\boxed{\boldsymbol{F}_{\text{tot}}(\mathbf{k}) = A(\mathbf{k})\boldsymbol{F}(\mathbf{k})} \qquad \text{(array pattern multiplication)} \tag{22.3.1}$$

where $A(\mathbf{k})$ is the *array factor*:

$$\boxed{A(\mathbf{k}) = a_0 e^{j\mathbf{k}\cdot\mathbf{d}_0} + a_1 e^{j\mathbf{k}\cdot\mathbf{d}_1} + a_2 e^{j\mathbf{k}\cdot\mathbf{d}_2} + \cdots} \qquad \text{(array factor)} \tag{22.3.2}$$

Since $\mathbf{k} = k\hat{\mathbf{r}}$, we may also denote the array factor as $A(\hat{\mathbf{r}})$ or $A(\theta, \phi)$. To summarize, the net effect of an array of identical antennas is to modify the single-antenna radiation vector by the array factor, which incorporates all the translational phase shifts and relative weighting coefficients of the array elements.

In [ ]: 1