UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA

Antenas

Métodos dos Momentos - Dipolos

Matheus Ferreira Bernardo de Souza

Prof: Odilon Maroja da Costa Pereira Filho

16/11/2021

1 - Objetivos

Análise numérica das propriedades de antenas tipo dipolos usando o Método dos Momentos.

2 - Geometria

Seja uma antena de dipolo linear, de comprimento L, ao longo do eixo dos z, alimentada pelo centro. Considere que o raio do fio a seja muito menor que o comprimento de onda, e que o metal da antena seja perfeitamente condutor.

3 - Prática

1. *Seguindo o procedimento mostrado em sala, obtenha os elementos das matrizes de Impedância (Z) e de Tensão(V) para funções de base tipo triangulares.*

Com o objetivo de obter os valores de impedância e corrente de uma dada antena (descrita na seção de geometria), de uma forma analitica o problema pode ser definido pelo seguinte sistema:

$$\boxed{[Z][I]=[V]}$$

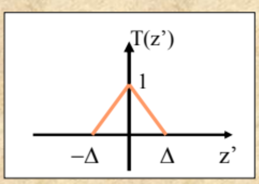
Onde:

$$\boxed{Z_{mn} = \int_{-l/2}^{l/2} W_m(z)\left(k^2 + \frac{\partial^2}{\partial z^2}\right) \int_{-l/2}^{l/2} U_n(z')\, \frac{e^{-jk\sqrt{(z-z')^2+a^2}}}{4\pi\sqrt{(z-z')^2+a^2}}\, dz'\, dz}$$

$$\boxed{V_m = -j\omega\varepsilon \int_{-l/2}^{l/2} W_m(z)\,\delta(z)\, dz = -j\omega\varepsilon\, W_m(0)}$$

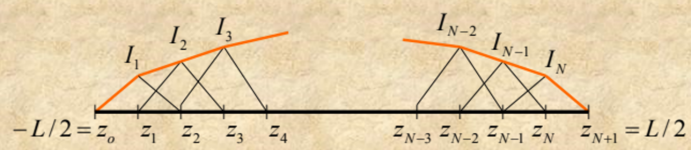
A corrente na antena não é conhecida, porém como sabemos de sua natureza comum de distribuição é possível fazer uma aproximação (chute educado) de como ela está distribuída ao longo do eixo Z da antena. Nessa atividade fazemos isso por meio de funções de base triangular:

$$\boxed{I_z(z') \approx \sum_{n=1}^{N} I_n U_n(z') = \sum_{n=1}^{N} I_n T(z'-z_n)}$$



$$z_n = -\tfrac{L}{2} + n\Delta \qquad \Delta = \frac{L}{N+1}$$

$$T(z') = 1 - \tfrac{|z'|}{\Delta} \qquad |z'| \leq \Delta$$



Por esse caminho podemos simplificar mais ainda alguns termos como:

$$\boxed{Z_{mn} = k^2 A_{mn} - \Phi_{mn}}$$

Onde:

$$\boxed{\Phi_{mn} \approx \Psi(m-1/2, n-1/2) - \Psi(m-1/2, n+1/2) - \Psi(m+1/2, n-1/2) + \Psi(m+1/2, n+1/2)}$$

$$A_{mn} = \int_{-l/2}^{l/2} W_m(z) \int_{-l/2}^{l/2} U_n(z')\; G(z-z')\, dz'\, dz =$$
$$\approx \Delta \int_{z_{n-1/2}}^{z_{n+1/2}} G(z_m - z')\, dz' = \Delta^2\; \Psi(m,n)$$

Sabendo que:

• **Aproximação para Ψ(m,n)**

i) Se m=n $\quad e^{-jk\sqrt{(z_m-z')^2+a^2}} \approx 1 - jk\sqrt{(z_m - z')^2 + a^2}$

$$\Psi(n,n) \approx \frac{1}{\Delta} \int_{z_{n-1/2}}^{z_{n+1/2}} \frac{1 - jk\sqrt{(z_n - z')^2 + a^2}}{4\pi\sqrt{(z_n - z')^2 + a^2}}\, dz' \approx \frac{1}{2\pi\Delta} \ln(\Delta/a) - \frac{jk}{4\pi}$$

ii) Se m ≠ n

$$\Psi(m,n) \approx \frac{e^{-jk\sqrt{(z_m - z_n)^2 + a^2}}}{4\pi\sqrt{(z_m - z_n)^2 + a^2}}$$

Dessa forma é possível implementar um software para fazer essa análise numérica do problema.

2. *Implemente a formulação em Matlab ou outro ambiente/linguagem desejada.*

A implementação foi feita usando a linguagem Python em conjunto com as bibliotecas scipy (para constantes matemáticas/físicas), numpy (para resolução de sistemas lineares) e matplotlib (para visualização dos resultados obtidos).

$$\boxed{[Z][I] = [V]}$$

```
# solve the linear system
I = np.linalg.solve(Z, V)
```

Onde:

$$Z_{mn} = k^2 A_{mn} - \Phi_{mn}$$

```
# populate impedance matrix
for m in range(self.N):
    for n in range(self.N):
        Z[m,n] = self.k**2 * self.__A(m,n) - self.__phi(m,n)
```

$$V_m = -j\omega\varepsilon \int_{-l/2}^{l/2} W_m(z)\,\delta(z)\,dz = -j\omega\varepsilon W_m(0)$$

```
# populate voltage matrix
V[round(((self.N+1)/2)-1), 0] = -self.j * self.w * self.e * self.volt
```

$$A_{mn} = \int_{-l/2}^{l/2} W_m(z) \int_{-l/2}^{l/2} U_n(z')\; G(z-z')\,dz'\,dz =$$
$$\approx \Delta \int_{z_{n-1/2}}^{z_{n+1/2}} G(z_m - z')\,dz' = \Delta^2\; \Psi(m,n)$$

```
def __A(self, m, n):
    return (self.delta**2 * self.__psi(m,n))
```

Se m = n:

$$\Psi(n,n) \approx \frac{1}{\Delta} \int_{z_{n-1/2}}^{z_{n+1/2}} \frac{1 - jk\sqrt{(z_n - z')^2 + a^2}}{4\pi\sqrt{(z_n - z')^2 + a^2}}\,dz' \approx \frac{1}{2\pi\Delta}\ln(\Delta/a) - \frac{jk}{4\pi}$$

Se m != n:

$$\Psi(m,n) \approx \frac{e^{-jk\sqrt{(z_m - z_n)^2 + a^2}}}{4\pi\sqrt{(z_m - z_n)^2 + a^2}}$$

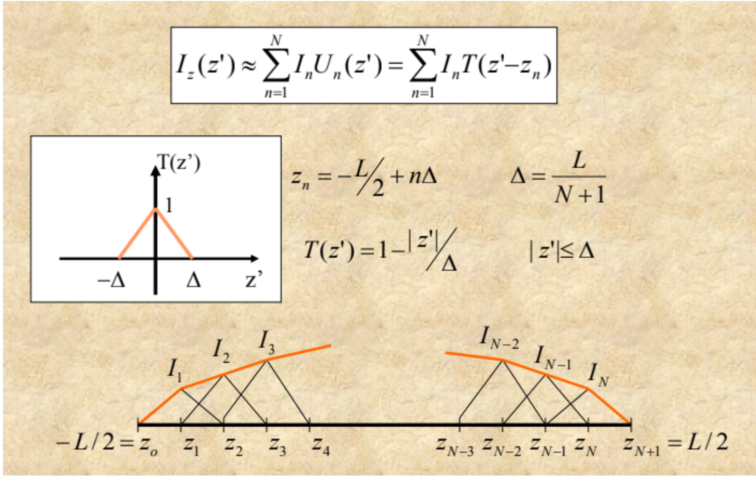
```
def __psi(self, m, n):
    if m == n:
        return ((1/(2*pi*self.delta))*np.log(self.delta/self.a) - self.j*self.k/(4*pi))
    else:
        return (self.__G(self.__z(m) - self.__z(n)))
```

```
def __G(self, z):
    return (np.exp(-self.j*self.k*np.sqrt(z**2 + self.a**2))/(4*pi*np.sqrt(z**2 + self.a**2)))
```

$$\Phi_{mn} \approx \Psi(m-1/2, n-1/2) - \Psi(m-1/2, n+1/2) - \Psi(m+1/2, n-1/2) + \Psi(m+1/2, n+1/2)$$

```
    def __phi(self, m, n):
        return ( self.__psi(m-0.5,n-0.5) - self.__psi(m-0.5,n+0.5) - self.__psi(m+0.5,n-0.5)
+ self.__psi(m+0.5,n+0.5) )
```

E finalmente as aproximações da corrente distribuída na antena:




```python
def __Triangular_Base(self, z):
    if abs(z) <= self.delta:
        return (1 - abs(z)/self.delta)
    else:
        return 0

def __z(self, n):
    return (-self.L/2 + n * self.delta)

def __Iz(self, I_array, z, base_function=None):
    i = 0.0 + 0.0j
    for n in range(self.N):
        i = i + I_array[n,0] * base_function(z - self.__z(n+1))
    return i
```

O código completo pode ser visto em anexo ao relatório.

3. *Para uma antena de meio comprimento-de-onda (L = λ/2), e raio a = 10^{-4}λ, determine a distribuição de corrente ao longo da antena. Use 19 funções de base. Varie o número de funções de base (N, ímpar) e observe a convergência. Lembre-se de que a alimentação deve ser mantida no elemento central (elemento (N+1)/2).*

```python
class Linear_Antenna:
    def __init__(self, frequency=300e6, radius=1e-4, lenght_factor=1/2, source_voltage=1) ->
 None:

        # antenna characteristics
        self.lbd = speed_of_light/frequency
        self.w = 2 * pi * frequency
        self.k = (2 * pi)/self.lbd
        self.a = radius * self.lbd
        self.L = self.lbd * lenght_factor
        self.volt = source_voltage
        self.delta = None
        self.N = None

        # constants
        self.j = 0 + 1j
        self.e = 8.854187817e-12
```

Na instanciação da antena foi definida uma frequência de 300 MHz, um raio de $10^{-4}\lambda$, uma excitação de 1V e um comprimento de onda de $\lambda/2$. Tudo de acordo com os requerimentos dados.

Resultados:

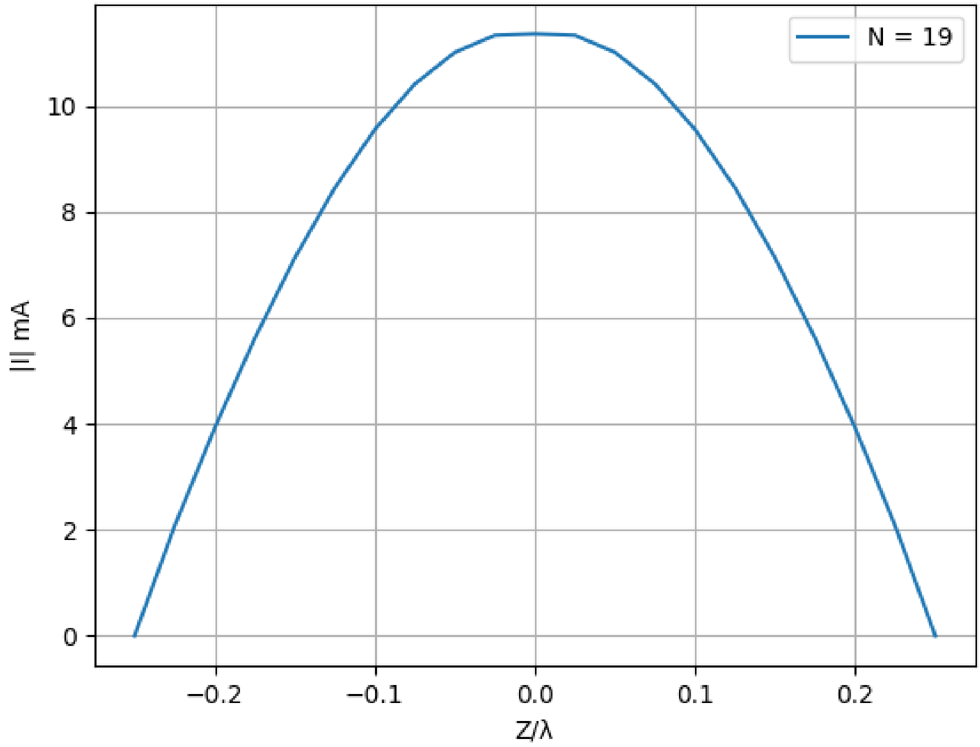


figura 1: distribuição da corrente na antena para 19 segmentos

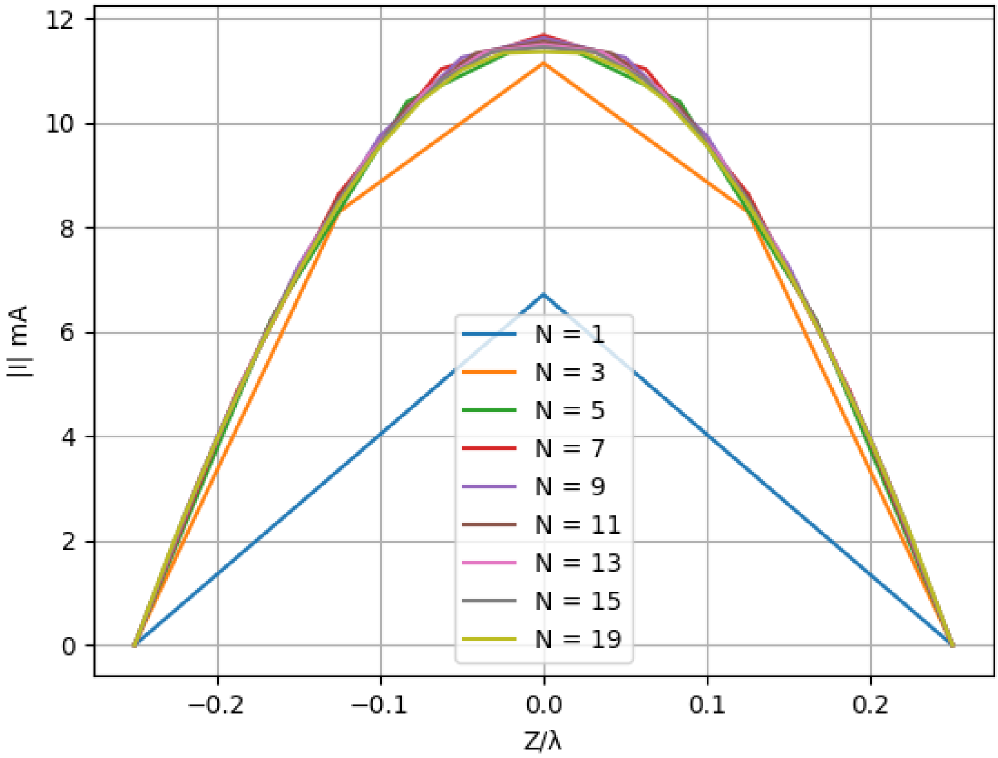


figura 2: distribuição da corrente na antena para 1 até 19 segmentos

__4.__ Para a antena do item anterior, obtenha a impedância de entrada, e mostre sua convergência com N.

Resultados:

O programa gera como saída as seguintes impedâncias para N = 1, 3, 5, 7, 9, 11, 13, 15, 19 respectivamente:

```
(52.183090696145946+139.5892368126632j)
(71.89350043622771+53.76754936284155j)
(75.12197828073145+41.374843489649784j)
(76.48748410676372+38.555314311735756j)
(77.25573604735224+37.96871891417739j)
(77.75445385864238+38.05113088506095j)
(78.10737835159556+38.349400438520455j)
(78.3719703472052+38.70860237848011j)
(78.74532474360599+39.415741281410476j)
```

Logo, para N = 19 o programa calculou uma impedância de 78.74532474360599 + 39.415741281410476j Ω

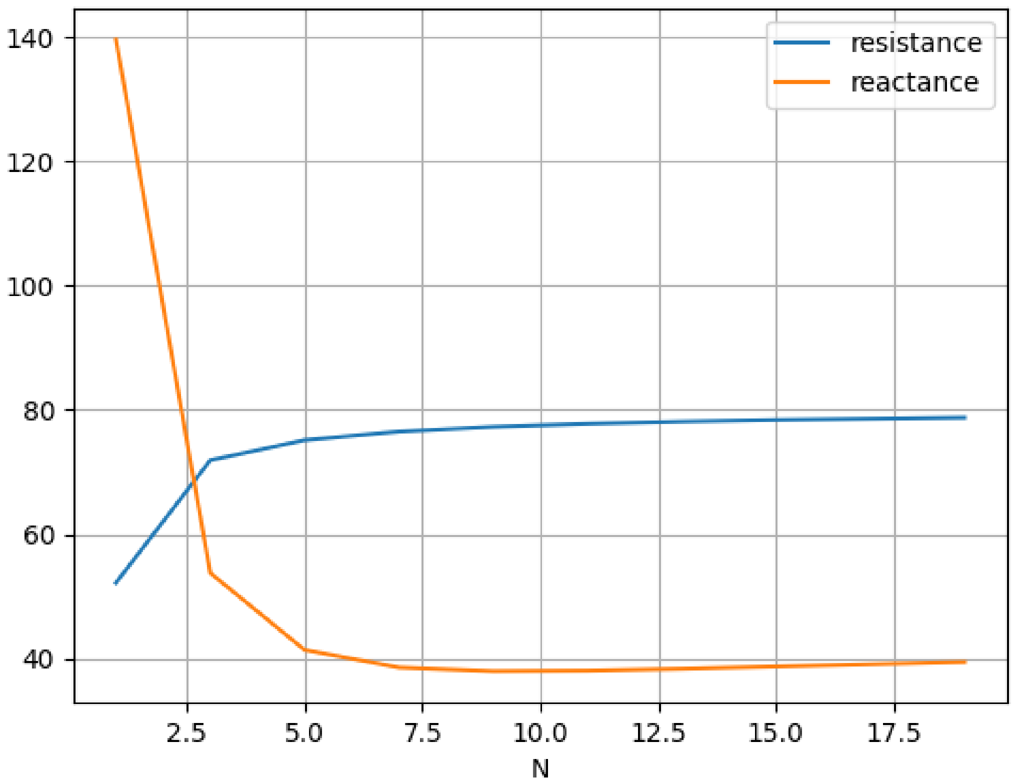


figura 3: gráfico da resistência e reatância à medida que N aumenta

Conclusões:

Com esse experimento foi possível aprender mais na prática os temas vistos em sala de aula sobre as naturezas físicas de uma antena dipolo linear, e aplicando os conceitos de álgebra linear e programação.