

Simulations of Dipole- and Array- Antenna in MATLAB

Chalmers University of Technology, Gothenburg, Sweden

by

Marika Svensson
Department of Signals and Systems

June 15, 2015

Properties of Sparse Array Antennas / by
Marika Svensson. - Gothenburg : Chalmers University of Technology, 2015.

Copyright ©2015 by Marika Svensson, Antenna Group, Signals and Systems Department,
Chalmers University of Technology, Gothenburg, Sweden; RUAG Space AB, Gotheburg,
Sweden.

This thesis is carried out under supervision of:
Dr. Johan Wettegren

Examiner:
Prof. Marianna Ivashina

Co-Supervisors:
Dr. Rob Maaskant
Carlo Bencivenni

Keywords: antenna arrays / irregular array antennas / ... / ...
Subject headings: antenna arrays / irregular array antennas / ... / ...

Technical Report No. 2015:xx
Antenna Group, Department of Signals & Systems
Chalmers University of Technology,
SE-41296, Gothenburg, Sweden.
Telephone: +46 31 772 1000

Cover design: Marika Svensson
Press: Chalmers Reproservice, Göteborg

The work presented in this thesis has been financed by ...; and has been performed at ...

Abstract

In this report two types of antennas were simulated in MATLAB. Initially a horizontal dipole antenna was investigated by computing the far field function, the radiated power and lastly the far field patterns in the E- and H-plane.

Then an irregularly spaced array antenna was investigated. First the far field was computed as a function of antenna positions and excitation current. Then the excitation current was computed as a function of scanning angles and array element positions. Lastly an equispaced array antenna was considered. The radiation pattern were computed for the H plane and broadside with different intermediate spacing d .

Contents

1	Task 1	2
1.1	Theory	2
1.2	Results	5
2	Task 2	7
2.1	Theory	7
2.2	Results	9
3	Task 3	10
3.1	Theory	10
3.2	Results	11
4	Task 4	13
4.1	Theory	13
4.2	Result	14

Contents	v
5 Task 5	16
5.1 Theory	16
5.2 Results	18
6 Task 6	20
6.1 Task 6 a	21
6.1.1 Theory	21
6.1.2 Results	21
6.2 Task 6 b	23
6.2.1 Theory	23
6.2.2 Results	23
7 Discussion and conclusion	27
A Source code	29
A.1 Task 1	29
A.1.1 Task1.m	29
A.1.2 horizontalDipole.m	32
A.2 Task 2	32
A.2.1 Task2.m	32
A.2.2 totalPwr.m	33

A.3	Task 3	34
A.3.1	Task3.m	34
A.3.2	totalPwr.m	35
A.4	Task 4	36
A.4.1	Task4.m	36
A.4.2	totalPwr.m	36
A.5	Task 5	38
A.5.1	Task5.m	38
A.5.2	excitationVector.m	39
A.5.3	excitationVector1.m	39
A.6	Task 6	40
A.6.1	Task6_a.m	40
A.6.2	Task6_b.m	41
A.6.3	totalPwr.m	43

Introduction

The purpose of this report was to evaluate the knowledge of antenna simulations and general competence in MATLAB. This was illustrated by computing the far field functions $\mathbf{G}(\theta, \phi)$, radiated power P_{rad} , radiation patterns, far field patterns, array factor patterns and excitation currents $\{I_n\}$ of different types of antennas in MATLAB.

Two kinds of antennas were considered. In chapters 1-3 a horizontal dipole antenna was simulated in MATLAB where first the far field of the dipole antenna was computed as a function of observation angles and the excitation current. Then the total radiated power was computed as a function of the far field, then the far field patterns and co polar radiation patterns evaluated as a function of the far field and observation angles in the E- and H-plane were computed and plotted.

In chapters 4- 6 the second type of antenna to be considered was array antennas. First the far field function was computed for an irregular array antenna as a function of element position and excitation current. Then the excitation current was computed, as a function of scanning angles and element positions for an irregularly spaced array antenna. The amplitude of this current was set to unity. Lastly an equispaced array was considered and the far field, the radiation patterns were computed in the H-plane and broadside as a function of the far field of the array antenna and observation angles.

In chapter 7 the results from the dipole and array antennas will be discussed, in appendix A the complete code written in MATLAB may be viewed by the reader.

Chapter 1

Task 1

In this section a horizontal dipole was simulated with MATLAB. The far field function of the dipole was computed as a function of the angles θ , ϕ and excitation current I , which was set to unity.

We will begin by presenting the theory of the dipole antenna and the far field region in electromagnetic theory. Then we will proceed with presenting the results generated from the MATLAB simulations.

1.1 Theory

Antenna theory can be greatly simplified by the fact that the fields and different quantities are considered in the far field region only, which is the region where $r > 2D^2/\lambda$. This means that many complicated integrals, Lagrange polynomials, sums etc. are instead approximated.

Begin by considering the electric field at a point in space $\mathbf{r} = \mathbf{R} + \mathbf{r}_0$ where the origin is set to be \mathbf{r}_0 for this coordinate system. The electric field can be considered as a function of R, θ, ϕ , i.e.

$$E(r, \theta, \phi) = E(R, \theta, \phi). \quad (1.1)$$

and the far field function times an exponent (which is the electrical field) can be rewritten

as

$$G(\theta, \phi) \frac{e^{-jkr}}{r} = G'(\theta, \phi) \frac{e^{-jkR}}{R} \approx \quad (1.2)$$

$$\approx G(\theta, \phi) e^{-jk(r-R)} \frac{e^{-jkR}}{R} = G(\theta, \phi) e^{-jk(\mathbf{r}_0 \cdot \hat{\mathbf{r}})} \frac{e^{-jkR}}{R} \quad (1.3)$$

where R is the distance $|\mathbf{r} - \mathbf{r}_0|$. In order to do the steps the Fraunhofer approximations in the far field were used,

$$\left\{ \begin{array}{l} \frac{1}{r} \approx \frac{1}{R} \\ r \approx R + \mathbf{r}_0 \cdot \hat{\mathbf{r}}. \end{array} \right. \quad (1.4)$$

It is obvious that the far field can be written as

$$\mathbf{G}_A(\theta, \phi) = \mathbf{G}(\theta, \phi) e^{jk\mathbf{r}_A \cdot \hat{\mathbf{r}}} \quad (1.5)$$

in a space position \mathbf{r}_A .

Consider now a dipole which has a \hat{I} direction, a corresponding current density to this dipole will be

$$\mathbf{J}_l(l') = I_0 j(l') \hat{I} \quad (1.6)$$

where I_0 is the magnitude of the current and j is set to

$$j(l) = \frac{\sin\left(k\left(\frac{l}{2} - |z'|\right)\right)}{\sin(kl/2)}. \quad (1.7)$$

The electric field for a dipole may be written as a

$$\mathbf{E}_d(\hat{\mathbf{r}}) = \frac{e^{-jkr}}{r} \mathbf{G}_d(\hat{\mathbf{r}}) \quad (1.8)$$

which is similar to equation (1.2) but with a far field which can be written as

$$\left\{ \begin{array}{l} G_d(\hat{\mathbf{r}}) = \eta I_0 C_k [\hat{I} - (\hat{I} \cdot \hat{\mathbf{r}}) \cdot \hat{\mathbf{r}}] \hat{j}(l\hat{I} \cdot \hat{\mathbf{r}}) \\ \hat{j} = \int_{-l/2}^{l/2} j(l') e^{klj' \hat{I} \cdot \hat{\mathbf{r}}} \approx l/2 \text{ if short dipole} \end{array} \right. \quad (1.9)$$

Consider finally a horizontal dipole, which is a dipole with its direction in the y direction. This gives the expressions

$$\hat{I} = \hat{y} \quad (1.10)$$

$$\hat{I} \cdot \hat{\mathbf{r}} = \sin\theta \sin\phi \quad (1.11)$$

$$\hat{I} - (\hat{I} \cdot \hat{\mathbf{r}}) \hat{\mathbf{r}} = \cos\theta \sin\phi \hat{\theta} + \cos\phi \hat{\phi}. \quad (1.12)$$

The equations above can be understood by considering the spherical coordinates

$$\begin{cases} \hat{x} = \sin\theta\cos\phi\hat{\theta} + \cos\theta\cos\phi\hat{\phi} - \sin\theta\hat{r} \\ \hat{y} = \sin\theta\sin\phi\hat{\theta} + \cos\theta\sin\phi\hat{\phi} - \sin\phi\hat{r} \\ \hat{z} = -\sin\theta\hat{\theta} - \cos\theta\hat{r}, \end{cases} \quad (1.13)$$

It can easily be seen that (1.12) indeed holds.

The far field function for a dipole of length l , sitting at height h can then be written as

$$\mathbf{G}(\theta, \phi) = C_k \eta I_0 \hat{j} (k \hat{I} \cdot \hat{r}) (\cos\theta \sin\phi \hat{\theta} + \cos\phi \hat{\phi}) (e^{jk h \cos\theta} - e^{-jk h \cos\theta}) \quad (1.14)$$

$$\approx C_k \eta I_0 \hat{j} (k \hat{I} \cdot \hat{r}) (\cos\theta \sin\phi \hat{\theta} + \cos\phi \hat{\phi}) (\sin(kh \cos(\theta))). \quad (1.15)$$

$$(1.16)$$

If the dipole is considered to be short $\hat{j} \approx l/2$, so the far field function can be described according to

$$\mathbf{G}(\theta, \phi) = G_\theta \hat{\theta} + G_\phi \hat{\phi}, \quad (1.17)$$

where the θ and ϕ parts of \mathbf{G} are given by the expression

$$G_\theta = G_\theta(\theta, \phi) = A \cos(\theta) \sin(\phi) 2j \sin(kh \cos(\theta)) \quad (1.18)$$

and

$$G_\phi = G_\phi(\theta, \phi) = A \cos(\phi) 2j \sin(kh \cos(\theta)). \quad (1.19)$$

The constant A is defined as

$$A = C_k \eta I_0 l / 2 \quad (1.20)$$

with

$$\begin{cases} C_k = -jk/4\pi \\ \eta \approx 120\pi [\Omega] \\ k = \frac{2\pi}{\lambda} \\ h = \frac{\lambda}{4} \\ I_0 = \text{incident current} \end{cases} \quad (1.21)$$

[1]

1.2 Results

The code for task 1 can be viewed in appendix A.1, where the length of the dipole was chosen to be 0.1λ and I to be 1 A in order to get numerical results.

The θ part of the far field function can be viewed in figure 1.1 as a function of the angles θ and ϕ .

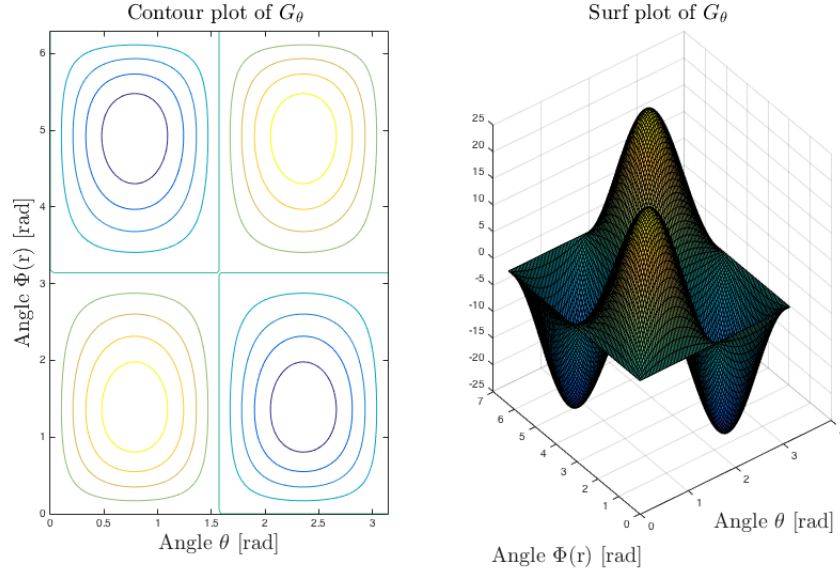


Figure 1.1: This figure shows G_θ as a function of θ and ϕ

The ϕ part of the far field function can be viewed in figure 1.2 as a function of the angles θ and ϕ .

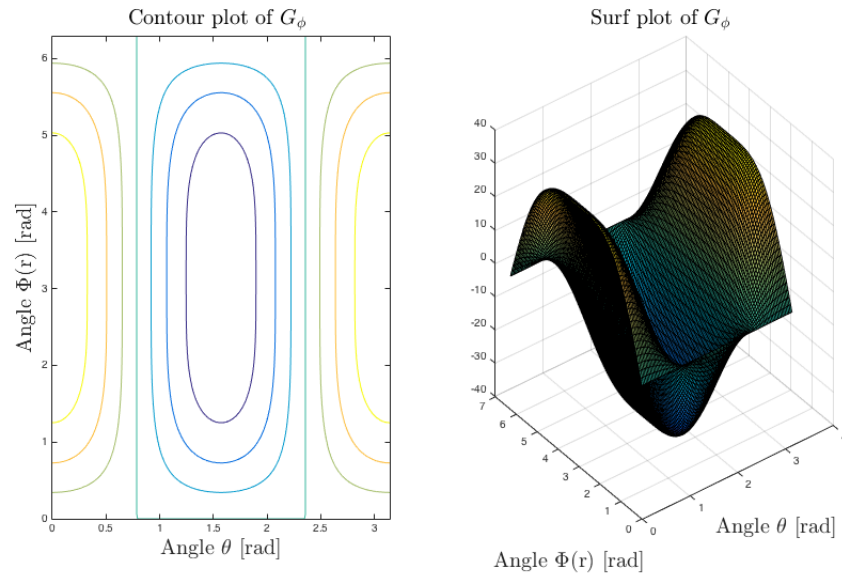


Figure 1.2: This figure shows G_ϕ as a function of θ and ϕ

Chapter 2

Task 2

In this chapter the total radiated power was computed as a function of the far field for the horizontal dipole, with a chosen polarization, that was considered in chapter 1. Initially the theory regarding the radiated power will be presented, then the results from the computations will be shown.

2.1 Theory

The total radiated power can be evaluated by considering the co- and cross-polarizations, which are defined as

$$\hat{c}o = \cos(\phi - \epsilon)\hat{\theta} - \sin(\phi - \epsilon)\hat{\phi} \quad (2.1)$$

and

$$\hat{x}p = -\sin(\phi - \epsilon)\hat{\theta} - \cos(\phi - \epsilon)\hat{\phi} \quad (2.2)$$

Where ϵ can be chosen to achieve the different polarizations. In order to achieve a linear x polarization we set $\epsilon = 0$ which gives us

$$\hat{c}o = \cos(\phi)\hat{\theta} - \sin(\phi)\hat{\phi} \quad (2.3)$$

$$\hat{c}o = -\sin(\phi)\hat{\theta} - \cos(\phi)\hat{\phi}. \quad (2.4)$$

To get the y polarization ϵ is set to $\pi/2$, giving the expression

$$\hat{c}o = \cos(\phi - \pi/2)\hat{\theta} - \sin(\phi - \pi/2)\hat{\phi} \quad (2.5)$$

$$\hat{c}o = -\sin(\phi - \pi/2)\hat{\theta} - \cos(\phi - \pi/2)\hat{\phi}. \quad (2.6)$$

For RHC we polarization vectors become

$$\hat{c}o = e^{-j\phi}/\sqrt{2}(\hat{\theta} - j\hat{\phi}) \quad (2.7)$$

$$\hat{x}p = e^{j\phi}/\sqrt{2}(\hat{\theta} + j\hat{\phi}) \quad (2.8)$$

and for LHC

$$\hat{c}o = e^{j\phi}/\sqrt{2}(\hat{\theta} + j\hat{\phi}) \quad (2.9)$$

$$\hat{x}p = e^{-j\phi}/\sqrt{2}(\hat{\theta} - j\hat{\phi}). \quad (2.10)$$

The total radiated power is defined to be

$$P_{rad} = \int \int_{\text{far field sphere}} (\langle \mathbf{W} \rangle \cdot \hat{r}) dA \quad (2.11)$$

which can be rewritten in spherical coordinates giving the expression

$$P_{rad} = \int_0^{2\pi} \int_0^\pi (\langle \mathbf{W} \rangle \cdot \hat{r}) r^2 \sin\theta d\theta d\phi. \quad (2.12)$$

The total radiated power can then be computed by integrating over the sphere of the squared norm of the far , i.e.

$$P_{rad} = \frac{1}{2\eta} \int_0^{2\pi} d\phi \int_0^\pi d\theta \sin(\theta) (|G_{co}|^2 + |G_{xp}|^2). \quad (2.13)$$

η can be set to $120\pi\Omega$ is room temperature. The integration can easily be calculated with the trapez method or the midpoint method in order to get a numerical result. In order to get the co- and cross- polar parts of the far field function we simply multiply it with the conjugate of the co- and cross- polar vectors, i.e.

$$G_{co} = \mathbf{G} \cdot \hat{c}o^* \quad (2.14)$$

$$G_{xp} = \mathbf{G} \cdot \hat{x}p^*. \quad (2.15)$$

Where we must choose a specific polarization in order to get the numerical results. [1]

2.2 Results

The MATLAB code for this task can be viewed in A.2. The numerical result of the total radiated power was 9.5984 W, with y, x, LHC and RHC polarization. The excitation current was chosen to be unity A for this task.

Chapter 3

Task 3

In this chapter the radiation and far field patterns in the H- and E-plane were computed in [dBi] as a function of the far field and observation angles in MATLAB. These patterns are useful to construct so that the radiation and far field functions from the antenna can be better understood. The radiation pattern is defined as the variation of radiated power by an antenna as a function of the direction from the antenna, the pattern is one of the quantities that are observed in the antenna's far field region which gives us the freedom to fully utilize the theory regarding the far field region and Fraunhofer approximation.

The theory is first presented in the section below and then the results from the simulations are presented in section 3.2.

3.1 Theory

For the horizontal dipole which is directed in the y direction the H-plane lies in the plane which corresponds to $\phi = 0$. The E-plane then must lie in the plane which has $\phi = \pi/2$, as the E-field is perpendicular to the H-field. With this knowledge it is easy to plot the radiation pattern for the co polarization according to the function

$$Pattern = 10 \cdot {}_{10}\log \left(\frac{4\pi |G_{co}(\theta, \phi_0)|^2}{P_{rad} 2\eta} \right). \quad (3.1)$$

Here the co polar radiation pattern is only considered, and is has also been normalized with respect to the radiated power. By taking the logarithm of the pattern the results will be given in [dBi] which is commonly used in antenna theory.

The far field patterns for the E- and H-plane of the far field function are defined as $|G_{co}(\theta, 0)|$ and $|G_{co}(\theta, \pi/2)|$ in the H- and E- planes respectively. These patterns can be plotted in dBi as

$$Pattern = 10 \cdot {}_{10}\log \left(\frac{4\pi |G_{co}(\theta, \phi_0)|}{\sqrt{P_{rad}2\eta}} \right). \quad (3.2)$$

[1]

3.2 Results

The MATLAB code for this task can be viewed in appendix A.3. The result for the co polarization radiation pattern in the E- and H-plane was plotted in figure 3.1 and 3.2, respectively.

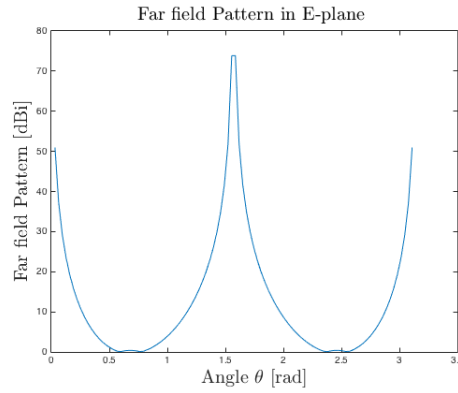


Figure 3.1: This figure shows the radiation pattern in the E-plane

The patterns of the far field function were plotted in figures 3.3 and 3.3 for the E- and H-plane.

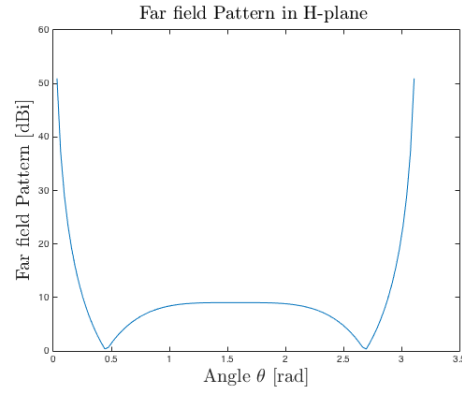


Figure 3.2: This figure shows the radiation pattern in the H-plane

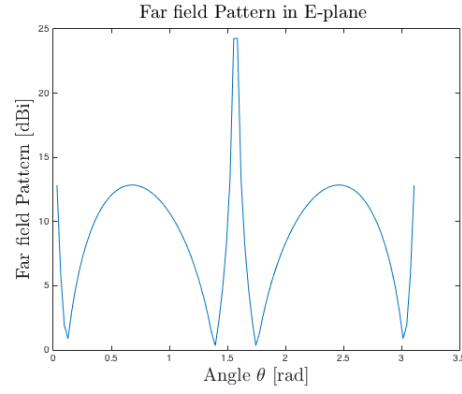


Figure 3.3: This figure shows the far field pattern in the E-plane

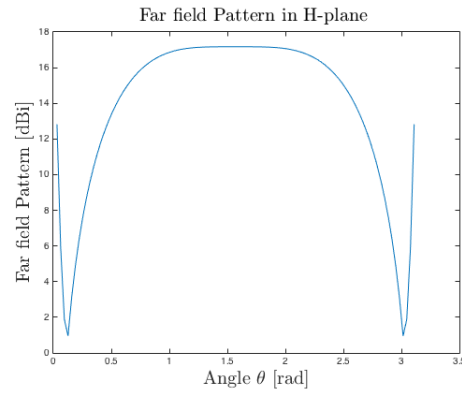


Figure 3.4: This figure shows the far field pattern in the H-plane

Chapter 4

Task 4

In this chapter we began to consider irregular array antennas with N identical elements. Array antennas are good to use as they can for example increase the overall gain, provide diversity reception, steer the array in a particular direction and determine the direction of arrival of the incoming signals. [2]

In this section specifically we were considering an array antenna with irregular spacing $\{\mathbf{r}_i\}_{i=1}^N$ and excitation currents $\{I_i\}_{i=1}^N$. A function was created that takes the positions, and excitation vector which gave the far field function of the array antenna as output.

4.1 Theory

For antennas with identical elements that lie in an arbitrary array the far field function may be computed as a sum over the elements in the array elements

$$\mathbf{G}_A(\theta, \phi) = \sum_{n=1}^N A_n e^{j\phi_n} (G_\theta \hat{\theta} + G_\phi \hat{\phi}) e^{jk\mathbf{r}_n \hat{r}}. \quad (4.1)$$

Which can be written as

$$\mathbf{G}_A(\theta, \phi) = \mathbf{G}(\theta, \phi) \cdot A\mathbf{F} \quad (4.2)$$

where AF stands for Array Factor. Further \mathbf{r}_n are the positions of the elements (assumed to be given in $m\lambda$), A_n and ϕ_n are the amplitude and phase of the current, respectively. The excitation current can be written in terms of amplitude and phase of the current

$$\mathbf{I} = [A_1 e^{j\Phi_1} A_2 e^{j\Phi_2} \dots A_{N-1} e^{j\Phi_{N-1}}], \quad (4.3)$$

it is then easy to compute the far field function. [1]

4.2 Result

The MATLAB code for this section can be viewed in appendix A.4. The absolute value of the θ and ϕ parts of the far field function were plotted in figures 4.1 and 4.2 with respect to the angles θ and ϕ , respectively.

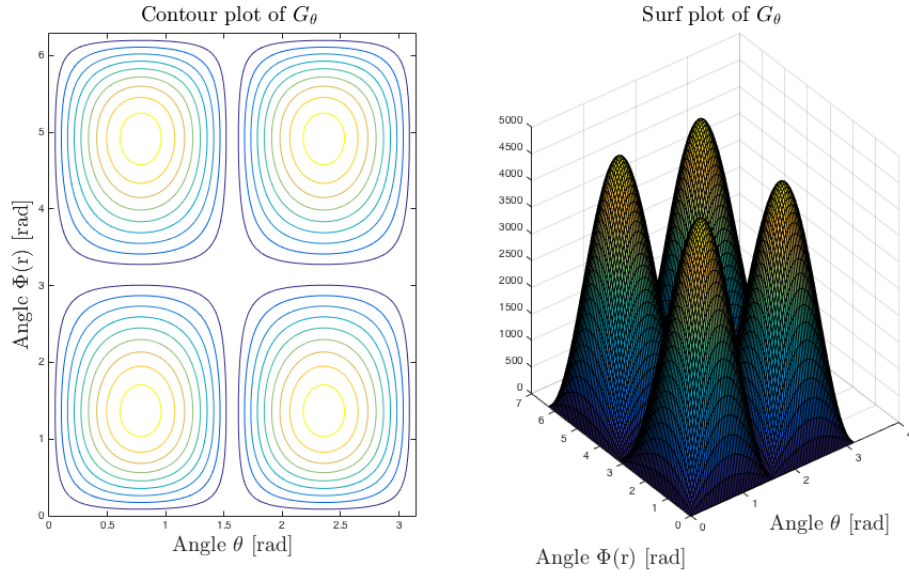


Figure 4.1: This figure shows G_θ as a function of θ and ϕ

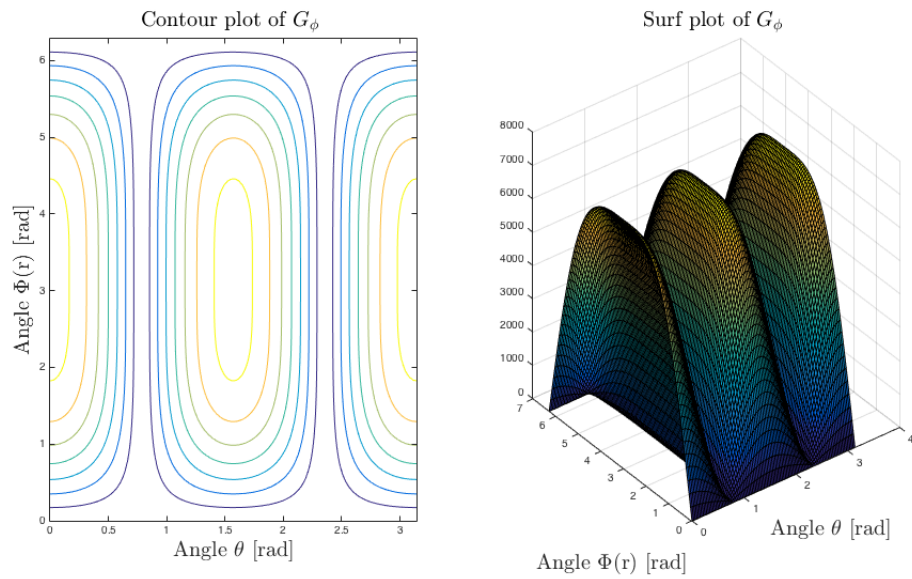


Figure 4.2: This figure shows G_ϕ as a function of θ and ϕ

Chapter 5

Task 5

In this chapter task 4 was continued, the aim was now to attempt to steer the irregular spaced array with an excitation current $I_n = A_n e^{j\Phi_n}$. This was illustrated by computing a MATLAB function which constructed the excitation current as a function of array antenna positions $\{r_i\}_{i=1}^N$ and steering angles θ_0, ϕ_0 .

To begin the theory regarding a steered array antenna and phase shifts will be presented. Then the results from the computations will be presented.

5.1 Theory

As mentioned before array antennas are good to use because they can be steered, i.e. every element is directed according to a steering vector with fixed angles θ_0 and ϕ_0 . The difference in phase (phase shift) between two elements are $\Phi_{n,n-1}$, which is the length difference of the two elements times the wave vector k . Consider now the spherical coordinates

$$\hat{r} = \sin\theta\cos\phi\hat{x} + \sin\theta\sin\phi\hat{y} + \cos\theta\hat{z}, \quad (5.1)$$

if the array antenna is steered in a direction θ_0 and ϕ_0 and the elements of the array are placed in positions $\{\mathbf{r}_i\}_{i=1}^N$ we may project the positions on the steering vector as

$$d_n = \mathbf{r}_n \cdot \hat{r} = (x_n\hat{x} + y_n\hat{y} + z_n\hat{z})(\sin\theta_0\cos\phi_0\hat{x} + \sin\theta_0\sin\phi_0\hat{y} + \cos\theta_0\hat{z}) = \quad (5.2)$$

$$\sin\theta_0\cos\phi_0x_n + \sin\theta_0\sin\phi_0y_n + \cos\theta_0z_n. \quad (5.3)$$

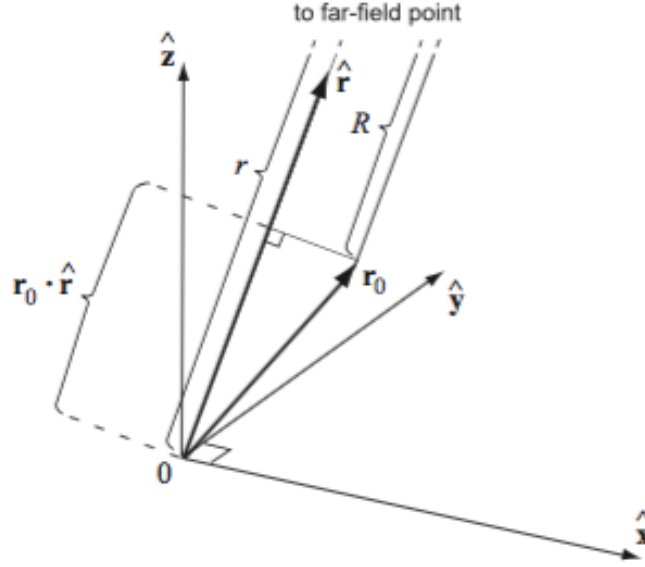


Figure 5.1: This figure shows how a vector can be projected onto another in the far field region of antennas. This figure was obtained from [1]

An illustration of the projection can be viewed in figure 5.1. The difference in length between two arbitrary elements in space is thus

$$d_{n,n-1} = |d_n - d_{n-1}| = |\mathbf{r}_n \cdot \hat{\mathbf{r}} - \mathbf{r}_{n-1} \cdot \hat{\mathbf{r}}| = \quad (5.4)$$

$$|\sin\theta_0 \cos\phi_0(x_n - x_{n-1}) + \sin\theta_0 \sin\phi_0(y_n - y_{n-1}) + \cos\theta_0(z_n - z_{n-1})|. \quad (5.5)$$

The phase difference between two elements subsequently be computed as

$$\Phi_{n,n-1} = kd_{n,n-1} = \frac{2\pi}{\lambda} |\sin\theta_0 \cos\phi_0(x_n - x_{n-1}) + \sin\theta_0 \sin\phi_0(y_n - y_{n-1}) + \cos\theta_0(z_n - z_{n-1})|. \quad (5.6)$$

However, the actual phase difference that is interesting to compute is the phase difference with respect to the steering vector for each element

$$\Phi_{n,0} = kd_{n,0} = \frac{2\pi}{\lambda} |\sin\theta_0 \cos\phi_0(x_n) + \sin\theta_0 \sin\phi_0(y_n) + \cos\theta_0(z_n)|. \quad (5.7)$$

It was assumed for this task that the amplitude of the current was 1 A and that the positions are given in $m\lambda$. Thus the excitation current can be computed to be

$$I_n = e^{j\Phi_{n,0}} = e^{\frac{2\pi}{\lambda} |\sin\theta_0 \cos\phi_0(x_n) + \sin\theta_0 \sin\phi_0(y_n) + \cos\theta_0(z_n)|}. \quad (5.8)$$

5.2 Results

The MATLAB scripts for this section are presented in section A.5. The result of the phase of the current was plotted in figure 5.2 for an equispaced and irregular array. The amplitude of the current was chosen not to be plotted since the amplitude of the excitation current was assumed to be unity.

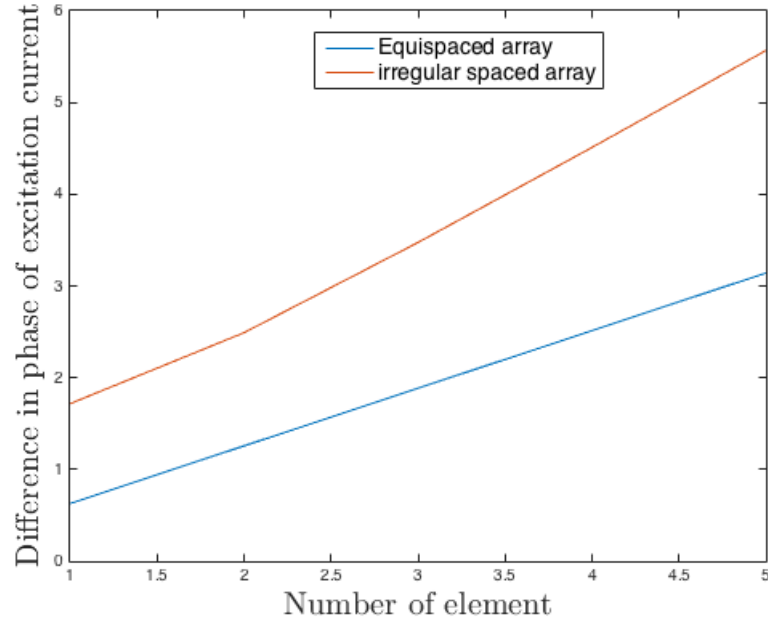


Figure 5.2: This figure shows the phase with respect to the steering vector of the excitation current as a function of element positions and steering direction $\theta_0 = \pi/3\text{rad}$ and $\phi_0 = \pi/2\text{rad}$

The phase difference between the elements were also computed and plotted in figure 5.3.

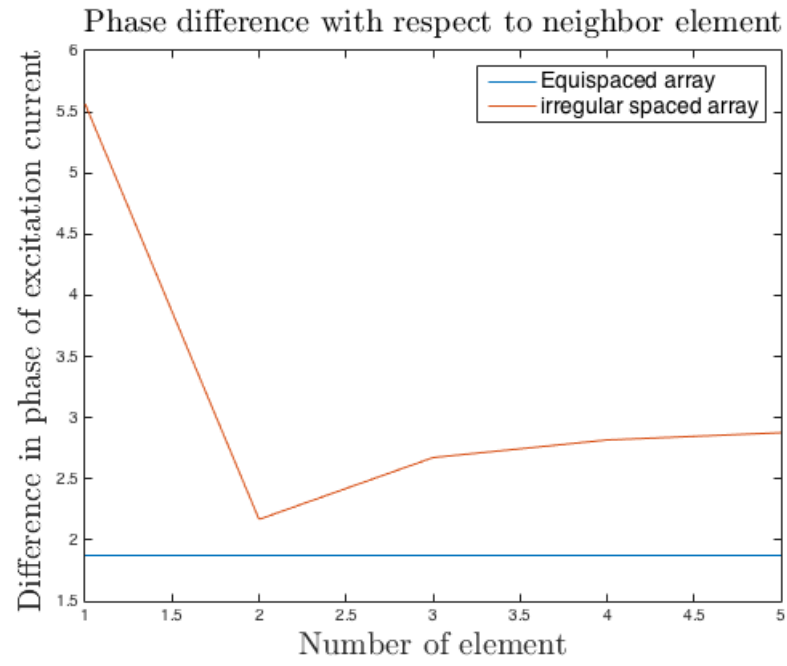


Figure 5.3: This figure shows the phase of the excitation current with respect to the neighbor as a function of element positions and steering direction $\theta_0 = \pi/3\text{rad}$ and $\phi_0 = \pi/2\text{rad}$

Chapter 6

Task 6

An N dimensional equispaced array antenna with identical elements was considered in this section. Having an equispaced array antenna simplifies the computations as we know that the phase difference between the elements will be the same for every element (i.e. constant phase difference).

The positions, $\{\mathbf{r}_n\}_{i=1}^N$, of the antennas can be written as

$$\mathbf{r}_n = \mathbf{r}_c + a_n \hat{a} = \mathbf{r}_c + \left(n + \frac{N+1}{2}\right) d \hat{a} \quad (6.1)$$

where \mathbf{r}_c is the center of the array and we have intermediate spacing d the total length is then Nd and we may write the electrical field as

$$\mathbf{E}_n(\mathbf{r}) = \frac{e^{-jkr}}{r} \mathbf{G}(\mathbf{r}). \quad (6.2)$$

We have earlier defined the far field function and the array factor in section 4, which now allows us to consider the array factors and the radiation patterns. If the antenna is directed at an angle θ the array factor can be written as

$$AF = 1 + e^{j(kd \cdot 1 \cos(\theta) + \alpha)} + \dots + e^{j \cdot (N-1)(kd \cos(\theta) + \alpha)} = \quad (6.3)$$

$$\sum_{n=1}^N e^{j(kd \cdot (n-1) \cos(\theta) + \alpha)} = e^{j(N-1)\Psi/2} \sin(N\Psi/2) / \sin(\Psi/2) \quad (6.4)$$

where $\Psi = \alpha + kd \cos(\theta)$ and assuming that the elements of the antenna are equal.

6.1 Task 6 a

In this section the array antenna with 5 elements was considered for intermediate element spacing $d = \lambda/4$ pointed in the H-plane. Firstly the theory regarding the array antennas and the H-plane will be presented. Then the results from the MATLAB simulations will be presented.

6.1.1 Theory

The total radiation pattern of an array antenna is defined as

$$Pattern_{radiation}^{tot} = (\text{element radiation pattern}) \times (AF). \quad (6.5)$$

For an equispaced array with intermediate spacing d and $N-1$ elements there will be a phase difference α between every element. The normalized array factor can be written as

$$|AF| = \sin(N\Psi/2)/(N\sin(\Psi/2)). \quad (6.6)$$

where $\Psi = \alpha + kdcos\theta$. As we are considering horizontal dipole antennas the radiation pattern in the far field region will be

$$f = \sqrt{1 - \sin^2(\theta)\cos^2(\phi)} \quad (6.7)$$

A maximum of the array factor can be found at $\Psi = 0$ which corresponds to $\alpha = kdcos(\theta_0)$ where θ_0 is called the steering angle. This will simplify the expression when the array antenna is directed at a specific region. The radiation patterns the total radiation pattern can thus be plotted in the H plane which corresponds with $\alpha = kdcos(\theta_0) = kd$, i.e. $\theta_0 = 0$ [2]. [1]

The radiation pattern, far field pattern may be normalized and plotted in dBi as described in chapter 3.

6.1.2 Results

The MATLAB scripts for the simulation of the H-plane can be viewed in appendix A.6.1. The plotted results for the radiation-, far field- and AF-patterns were plotted in figures

6.1, 6.2 and 6.3, respectively.

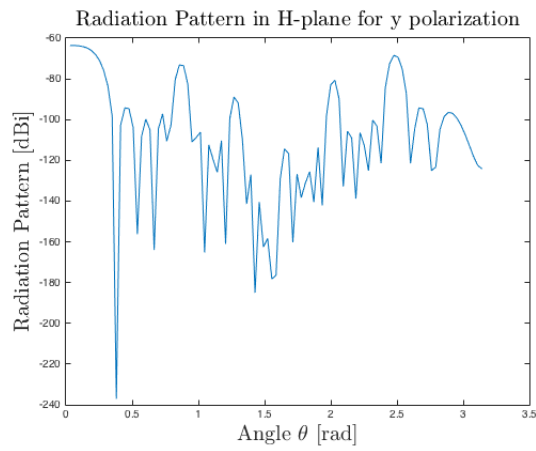


Figure 6.1: This figure shows the radiation pattern as a function of θ in the H-plane

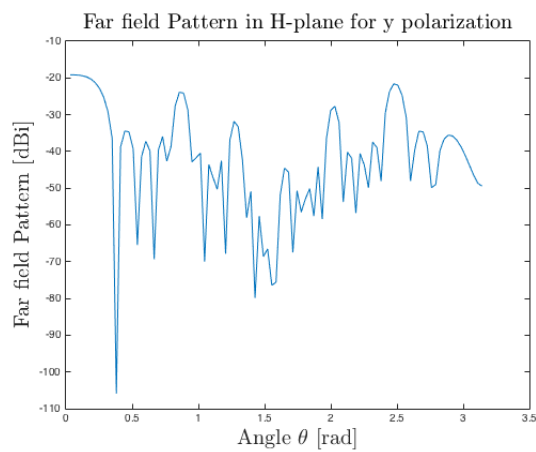


Figure 6.2: This figure shows the far field pattern as a function of θ in the H-plane

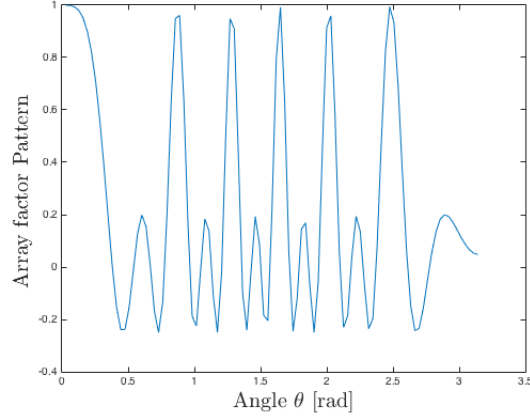


Figure 6.3: This figure shows AF as a function of θ and ϕ for y polarized dipoles

6.2 Task 6 b

In this section the array antenna with 5 elements was considered for intermediate element spacing $d \in \lambda[1/4, 4]$ pointed in the broadside direction. Firstly the theory regarding the array antennas and the broadside will be presented. Then the results from the MATLAB simulations will be presented.

6.2.1 Theory

Consider the theory in task 6a, this is the theory which is needed to compute the result here as well but with some small adjustments. The radiation patterns the total radiation pattern can be plotted in the broadside, this corresponds with $\alpha = kdcos(\theta_0) = 0$. The steering angle is thus $\theta_0 = \pi/2$. [1]

6.2.2 Results

The MATLAB scripts for the simulation if the H-plane can be viewed in appendix A.6.1. The plotted results for the radiation-, far field- and AF-patterns were plotted in figures 6.4, 6.5 and 6.6, respectively. The figures illustrates the radiation pattern for $d \in \lambda[1/4, 4]$.

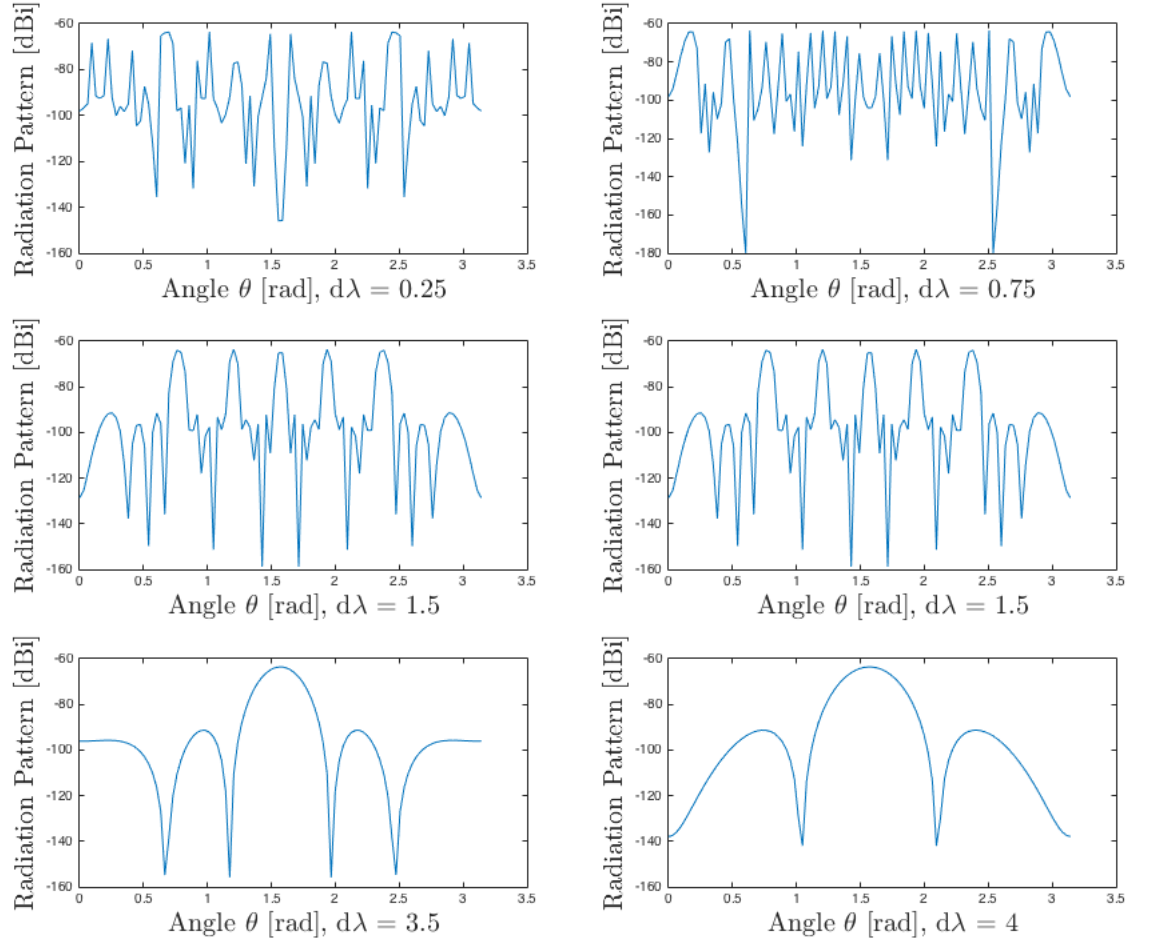


Figure 6.4: This figure shows the radiation pattern as a function of θ for y polarized dipoles in an array

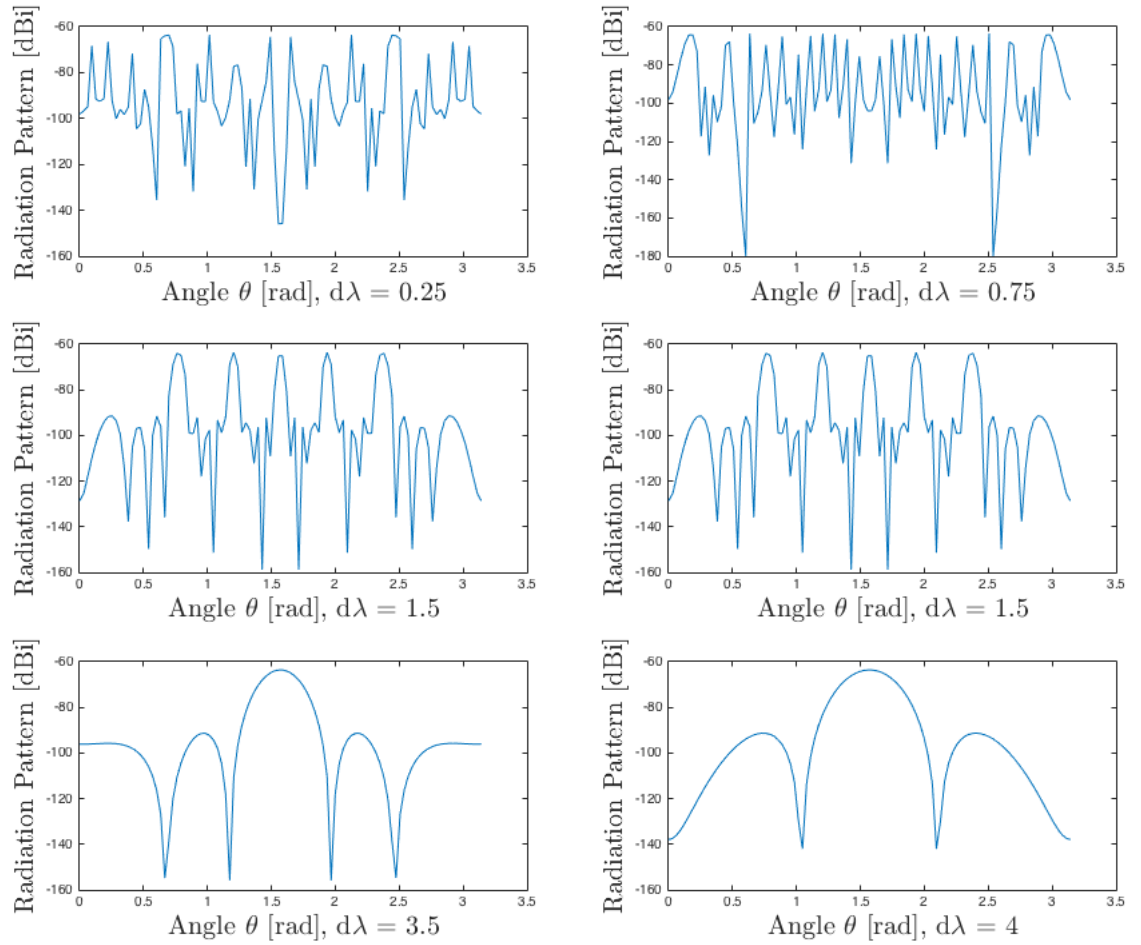


Figure 6.5: This figure shows the far field pattern as a function of θ for y polarized dipoles in arrays

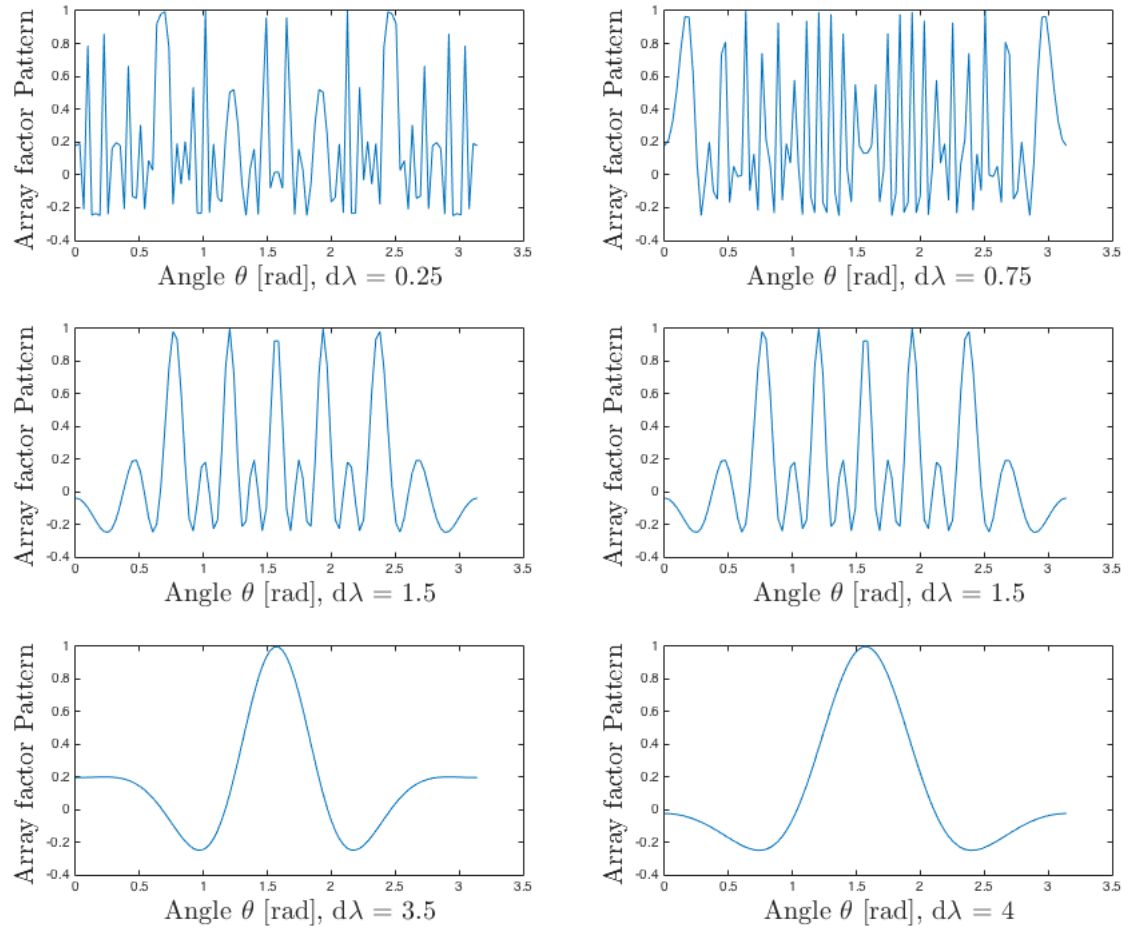


Figure 6.6: This figure shows AF as a function of θ and ϕ for y polarized dipoles

Chapter 7

Discussion and conclusion

In the first task the different parts of the far field function could be viewed in the plots, I have not found any reference for this function yet, but it appear to correspond to the analytical expression. Thus the conclusion is that the MATLAB function works correctly.

In task 2 the result of the total radiation the result gave the same for every polarization, as expected.

In task 3 the patterns of the far field function were plotted when normalized with respect to the total radiated power in the E- and H-plane. These patterns do resemble the expected result.

In the fourth task the different parts of the far field function could be viewed in the plots for a given current I , as a arbitrary current was chosen it was somewhat difficult to analyze the result. The result resembles the result in task 1 somewhat. Thus the conclusion is that the MATLAB function works correctly.

In task 5 it can be noted that the MATLAB function gives the expected phase difference for the equispaced array as the phase linearly increases with the distance from the origin. Simultaneously the phase difference between the individual neighbor elements remained constant through out the array.

In task 6 the patterns are quite symmetric, which was expected. Giving the conclusion

that the MATLAB function is correct.

Appendix A

Source code

A.1 Task 1

A.1.1 Task1.m

```
1 % Task 1
2 clear;
3 clc;
4 clf;
5 th = linspace(0,pi);
6 phi = linspace(0,2*pi);
7
8
9 for i=1:length(th)
10     for j=1:length(phi)
11
12         [Gth(i,j)  Gphi(i,j)] = horizontalDipole( 1, th(i), phi(j));
13     end
14 end
15
16
17 figure(1)
18 subplot(1,2,1)
19 contour(th, phi, Gth);
20 xlabel( Angle $\theta$ [rad] , Interpreter , latex , Fontsize ,20)
21 ylabel( Angle $\Phi$(r) [rad] , Interpreter , latex , Fontsize ,20)
22 title( Contour plot of $G_{\theta}$ , Interpreter , latex , Fontsize ,20)
23
24 subplot(1,2,2)
```

```

25 surf(th, phi, Gth);
26 xlabel( Angle $\theta$ [rad] , Interpreter , latex , Fontsize ,20)
27 ylabel( Angle $\Phi$(r) [rad] , Interpreter , latex , Fontsize ,20)
28 title( Surf plot of $G_{\theta}$ , Interpreter , latex , Fontsize ,20)
29
30
31 figure(2)
32 subplot(1,2,1)
33 contour(th, phi, Gphi);
34 xlabel( Angle $\theta$ [rad] , Interpreter , latex , Fontsize ,20)
35 ylabel( Angle $\Phi$(r) [rad] , Interpreter , latex , Fontsize ,20)
36 title( Contour plot of $G_{\phi}$ , Interpreter , latex , Fontsize ,20)
37
38
39
40 subplot(1,2,2)
41 surf(th, phi, Gphi);
42 xlabel( Angle $\theta$ [rad] , Interpreter , latex , Fontsize ,20)
43 ylabel( Angle $\Phi$(r) [rad] , Interpreter , latex , Fontsize ,20)
44 title( Surf plot of $G_{\phi}$ , Interpreter , latex , Fontsize ,20)
45
46 %%
47 clear;
48 clc;
49 clf;
50 u = linspace(-1,1);
51 v = linspace(-1,1);
52
53
54 for i=1:length(u)
55     for j=1:length(v)
56
57         [Gth(i,j)  Gphi(i,j)] = uvHorizontalDipole( 1, u(i), v(j));
58     end
59 end
60
61
62 figure(1)
63 subplot(1,2,1)
64 contour(u, v, Gth);
65 xlabel( $u = \sin\theta\sin\phi$ , Interpreter , latex , Fontsize ,20)
66 ylabel( $v = \sin\theta\sin\phi$ , Interpreter , latex , Fontsize ,20)
67 title( Contour plot of $G_{\theta}$ , Interpreter , latex , Fontsize ,20)
68
69 subplot(1,2,2)
70 surf(u, v, Gth);
71 xlabel( $u = \sin\theta\sin\phi$ , Interpreter , latex , Fontsize ,20)
72 ylabel( $v = \sin\theta\sin\phi$ , Interpreter , latex , Fontsize ,20)
73 title( Surf plot of $G_{\theta}$ , Interpreter , latex , Fontsize ,20)
74
75
76 figure(2)
77 subplot(1,2,1)
78 contour(u, v, Gphi);

```

```

79 xlabel( $u = \sin\theta\sin\phi$ , Interpreter , latex , Fontsize ,20)
80 ylabel( $v = \sin\theta\sin\phi$ , Interpreter , latex , Fontsize ,20)
81 title( Contour plot of $G_{\phi}$ , Interpreter , latex , Fontsize ,20)
82
83
84
85 subplot(1,2,2)
86 surf(u, v, Gphi);
87 xlabel( $u = \sin\theta\sin\phi$ , Interpreter , latex , Fontsize ,20)
88 ylabel( $v = \sin\theta\sin\phi$ , Interpreter , latex , Fontsize ,20)
89 title( Surf plot of $G_{\phi}$ , Interpreter , latex , Fontsize ,20)
90
91
92
93
94
95
96
97
98
99
100
101
102 %%
103 clear;
104 clc;
105 % define parameters
106
107 % th in [0,pi]
108 % phi in [0,2pi]
109 lambda =1;
110 k = 4*pi/lambda;
111 eta = 120*pi;
112 Ck = -i*k/(4*pi);
113 l = lambda / 2;
114 h = lambda/4;
115
116
117 % define theta part of G
118 G_th = @( th, phi, I) k/(2*pi)*eta*abs(I)*(cos(th)*sin(phi))*sin(pi/2*sin(th));
119
120 % define phi part of G
121 G_phi = @( th, phi, I) k/(2*pi)*eta*abs(I) *(cos(phi))*sin(pi/2*sin(th));
122
123
124 th = linspace(0,pi);
125 phi = linspace(0,2*pi);
126
127
128 for i=1:length(th)
129     for j=1:length(phi)
130         Gphi(i,j)= G_phi(th(i), phi(j), 1);
131
132         Gth(i,j) = G_th( th(i), phi(j), 1);

```

```

133         end
134     end
135     figure(1)
136     subplot(2,1,1)
137     contour(th, phi, Gth);
138
139     subplot(2,1,2)
140     surf(th, phi, Gth);
141
142     %%
143     figure(2)
144     subplot(2,1,1)
145     contour(th, phi, Gphi);
146
147     subplot(2,1,2)
148     surf(th, phi, Gphi);

```

A.1.2 horizontalDipole.m

```

1 function [ Gth Gphi ] = horizontalDipole( I, th, phi )
2 %UNTITLED13 Summary of this function goes here
3 % Detailed explanation goes here
4
5 % define theta part of G
6 Gth = 12*pi*abs(I) *(cos(th)*sin(phi))*sin(pi/2*sin(th));
7
8 % define phi part of G
9 Gphi = 12*pi*abs(I) *(cos(phi))*sin(pi/2*sin(th));
10
11 end

```

A.2 Task 2

A.2.1 Task2.m

```

1 clear; clc;
2 I =1; % set the current
3
4 % define theta part of G
5 G_th = @(th, phi,I) 12*pi*abs(I) *(cos(th)*sin(phi))*sin(pi/2*sin(th));

```

```

6
7 % define phi part of G
8 G_phi = @(th,phi, I) 12*pi*abs(I) *(cos(phi))*sin(pi/2*sin(th));
9
10 % define total G
11 G = @(th, phi) [G_th(th, phi,1) G_phi(th,phi,1)];
12 e=pi/2;
13 Prad= totalPwr(G, e)

```

A.2.2 totalPwr.m

```

1 function [ Prad ] = totalPwr(G,e)
2 %UNTITLED15 Summary of this function goes here
3 % Detailed explanation goes here
4 % G = [Gth(th, phi) Gphi(th, phi)] is a function
5
6 % Polarization:
7 %e = pi/2;
8 %Phi = pi/4;
9
10 % vectors for integration
11 phi = linspace(0,2*pi);
12 th = linspace(0,pi);
13 for n=1:length(phi)
14     costar = [conj(cos(phi(n)-e))
15              conj(-sin(phi(n)-e))];
16     xpstar = [conj(-sin(phi(n)-e))
17              conj(-cos(phi(n)-e))];
18     costar2 = e^(j*phi(n))*[1 -j]/sqrt(2);
19     costar3 = e^(-j*phi(n))*[1 +j]/sqrt(2);
20     xpstar2 = e^(j*phi(n))*[1 -j]/sqrt(2);
21     xpstar3 = e^(-j*phi(n))*[1 +j]/sqrt(2);
22     for m = 1:length(th)
23         Gco = G(th(m),phi(n))*costar3 ;
24         Gxp = G(th(m),phi(n))*xpstar3 ;
25         U(n,m) = abs(Gco)^2 + abs(Gxp)^2;
26
27     end
28 end
29
30 Prad =0;
31 h= (2*pi^2/(length(phi)*length(th)));
32 % integration with midpoint method.
33 for n = 1:length(th)
34     for m = 1:length(phi)
35         Prad = Prad + U(n,m)*sin(th(n))*h;
36     end
37 end
38 Prad = Prad/(2*120*pi);

```

39 `end`

A.3 Task 3

A.3.1 Task3.m

```

1 % task 3, prog 2: is the best program so far
2 clear; clc; close all;
3
4 I =1; % set the current
5
6 % define theta part of G
7 G_th = @(th, phi) 12*pi*abs(I) *(cos(th)*sin(phi))*sin(pi/2*sin(th));
8
9 % define phi part of G
10 G_phi = @(th,phi) 12*pi*abs(I) *(cos(phi))*sin(pi/2*sin(th));
11
12 % define total G
13 G = @(th, phi) [G_th(th, phi) G_phi(th,phi)];
14
15 % Polarization:
16 e = pi/2;
17
18 costar = @(phi) [conj(cos(phi-e))
19                 conj(-sin(phi-e))];
20 xpstar = @(phi) [conj(-sin(phi-e))
21                 conj(-cos(phi-e))];
22 Gco = @(th, phi ) G(th,phi)*costar(phi);
23 D = @(th, phi, P) 10*log(4*pi/(P)* norm(Gco(th,phi)^2));
24
25 % Eplane phi = pi/2
26 theta = linspace(0,pi);
27 P= totalPwr(G);
28
29 for n = 1:length(theta)
30     Dir(n) = abs(D(theta(n),pi/2, P ));
31 end
32
33 figure(1)
34 plot(theta(1:length(theta)-1), Dir(1:length(theta)-1))
35 xlabel( Angle $\theta$ [rad] , Interpreter , latex , Fontsize ,20)
36 ylabel( Far field Pattern [dBi] , Interpreter , latex , Fontsize ,20)
37 title( Far field Pattern in E-plane , Interpreter , latex , Fontsize ,20)
38
39 % Hplane : phi = 0;
40 theta = linspace(0,pi);

```



```

41 P= totalPwr(G);
42
43 for n = 1:length(theta)
44     Dir(n) = abs(D(theta(n),0, P ));
45 end
46
47 figure(2)
48 plot(theta(1:length(theta)-1), Dir(1:length(theta)-1))
49 xlabel( Angle $\theta$ [rad] , Interpreter , latex , Fontsize ,20)
50 ylabel( Far field Pattern [dBi] , Interpreter , latex , Fontsize ,20)
51 title( Far field Pattern in H-plane , Interpreter , latex , Fontsize ,20)

```

A.3.2 totalPwr.m

```

1 function [ Prad ] = totalPwr(G)
2 %UNTITLED15 Summary of this function goes here
3 % Detailed explanation goes here
4 % G = [Gth(th, phi) Gphi(th, phi)] is a function
5
6 % Polarization:
7 e = pi/2;
8
9
10 % vectors for integration
11 phi = linspace(0,2*pi);
12 th = linspace(0,pi);
13
14 for n=1:length(phi)
15     costar = [conj(cos(phi(n)-e))
16              conj(-sin(phi(n)-e))];
17     xpstar = [conj(-sin(phi(n)-e))
18              conj(-cos(phi(n)-e))];
19     for m = 1:length(th)
20         Gco = G(th(m),phi(n))*costar;
21         Gxp = G(th(m),phi(n))*xpstar;
22         U(n,m) = norm(Gco)^2 + norm(Gxp)^2;
23
24     end
25 end
26
27 Prad =0;
28 h= (2*pi^2/(length(phi)*length(th)));
29 % integration with trapez method.
30 for n = 1:length(th)
31     for m = 1:length(phi)
32         Prad = Prad + U(n,m)*sin(th(n))*h;
33     end
34 end
35

```

36 `end`

A.4 Task 4

A.4.1 Task4.m

```

1 clear;
2 clc;
3 clf;
4 %Initialize position matrix
5 for n= 1:10
6     A(n,:) = [n n+1 n-1];
7     I(n)= 1*exp(j*2*pi/n);
8 end
9
10 G = compGfromAandI(A,I);

```

A.4.2 totalPwr.m

```

1 function [ G ] = compGfromAandI( A, I )
2 %UNTITLED9 Summary of this function goes here
3 % Detailed explanation goes here
4
5 for n=1:length(I)
6     Amplitude(n) = norm(I);
7     phase(n) = atan(imag(I)/real(I));
8 end
9 th = linspace(0,pi);
10 phi = linspace(0,2*pi);
11
12 factor = @(th,phi, p) Amplitude(p)*exp(j*phase(p))*exp(j*2*pi*(A(1)*sin(th)*cos(phi)+ A(2)*sin(
    (th)*sin(phi)...
13         + A(3)*cos(th)));
14 % define theta part of G
15 G_th = @(th, phi, current) 12*pi*abs(current) *(cos(th)*sin(phi))*sin(pi/2*sin(th));
16
17 % define phi part of G
18 G_phi = @(th,phi, current) 12*pi*abs(current) *(cos(phi))*sin(pi/2*sin(th));
19
20 Gath = zeros(length(th), length(phi));

```

```

21 GAphi = zeros(length(th), length(phi));
22 for n=1:length(th)
23     for m=1:length(phi)
24         for p=1:length(I) % sum over elements
25
26             F = factor(th(n), phi(m), p);
27
28             GAphi(n,m) = GAphi(n,m) + F * G_phi(th(n), phi(m), I(1));
29
30             Gath(n,m) = Gath(n,m) + F * G_th(th(n), phi(m), I(1));
31         end
32     end
33 end
34
35 G=[Gath GAphi];
36
37 figure(1)
38 subplot(1,2,1)
39 contour(th, phi, abs(Gath));
40 xlabel( Angle $\theta$ [rad] , Interpreter , latex , Fontsize ,20)
41 ylabel( Angle $\Phi$(r) [rad] , Interpreter , latex , Fontsize ,20)
42 title( Contour plot of $G_{\theta}$ , Interpreter , latex , Fontsize ,20)
43
44 subplot(1,2,2)
45 surf(th, phi, abs(Gath));
46 xlabel( Angle $\theta$ [rad] , Interpreter , latex , Fontsize ,20)
47 ylabel( Angle $\Phi$(r) [rad] , Interpreter , latex , Fontsize ,20)
48 title( Surf plot of $G_{\theta}$ , Interpreter , latex , Fontsize ,20)
49
50 figure(2)
51 subplot(1,2,1)
52 contour(th, phi, abs(GAphi));
53 xlabel( Angle $\theta$ [rad] , Interpreter , latex , Fontsize ,20)
54 ylabel( Angle $\Phi$(r) [rad] , Interpreter , latex , Fontsize ,20)
55 title( Contour plot of $G_{\phi}$ , Interpreter , latex , Fontsize ,20)
56
57
58
59 subplot(1,2,2)
60 surf(th, phi, abs(GAphi));
61 xlabel( Angle $\theta$ [rad] , Interpreter , latex , Fontsize ,20)
62 ylabel( Angle $\Phi$(r) [rad] , Interpreter , latex , Fontsize ,20)
63 title( Surf plot of $G_{\phi}$ , Interpreter , latex , Fontsize ,20)
64
65
66
67
68 end

```

A.5 Task 5

A.5.1 Task5.m

```

1  clear;
2  clc;
3  close all;
4
5  % steering angles
6  th_0 = pi/3;
7  phi_0 = pi/2;
8
9
10 max = 5;
11 A1 = zeros(max,3);
12 for n = 1:max
13     elements(n) = n;
14     A1(n, 3) = n/max;
15 end
16 [CurrentEquispaced psi1] = excitationVector1( th_0, phi_0 , A1);
17
18 A2 = zeros(max,3);
19 for n = 1:max
20     A2(n, 1) = n^2/max;
21     A2(n, 2) = n/max ;
22     A2(n, 3) = 1/n/max ;
23 end
24 [CurrentNonEquispaced psi2] = excitationVector1( th_0, phi_0 , A2);
25 figure(2)
26 plot(elements, psi1, elements, psi2);
27 h_legend = legend( Equispaced array , irregular spaced array )
28 set(h_legend, FontSize ,14);
29 xlabel( Number of element , Interpreter , latex , Fontsize ,20)
30 ylabel( Difference in phase of excitation current , Interpreter , latex , Fontsize ,20)
31
32
33 clear;
34 clc;
35 close all;
36
37 % steering angles
38 th_0 = pi/3;
39 phi_0 = pi/2;
40
41
42 max = 5;
43 A1 = zeros(max,3);
44 for n = 1:max
45     elements(n) = n;
46     A1(n, 3) = n/max;

```

```

47 end
48 [CurrentEquispaced psi1] = excitationVector( th_0, phi_0 , A1);
49
50 A2 = zeros(max,3);
51 for n = 1:max
52     A2(n, 1) =n^2/max;
53     A2(n, 2) =n/max ;
54     A2(n, 3) =1/n/max ;
55 end
56 [CurrentNonEquispaced psi2] = excitationVector( th_0, phi_0 , A2);
57 figure(2)
58 plot(elements, psi1, elements, psi2);
59 h_legend = legend( Equispaced array , irregular spaced array )
60 set(h_legend, FontSize ,14);
61 xlabel( Number of element , Interpreter , latex , Fontsize ,20)
62 ylabel( Difference in phase of excitation current , Interpreter , latex , Fontsize ,20)
63 title( Phase difference with respect to neighbor element , Interpreter , latex , Fontsize ,20)

```

A.5.2 excitationVector.m

```

1 function [ I , phase] = excitationVector( th, phi , A)
2 %UNTITLED4 Summary of this function goes here
3 % Detailed explanation goes here
4 I = zeros(length(A(:,1)),1);
5 Psi = zeros(length(A(:,1)),1);
6 Psi(1) = exp(2*pi *abs(sin(th)*cos(phi)*(A(1,1))...
7             + sin(th)*sin(phi)*(A(1,2)) + cos(th)*(A(1,3)) ) );
8 I(1) = exp(j*Psi(1));
9 for n = 2:length(A(:,1))
10     Psi(n) = exp(2*pi *abs(sin(th)*cos(phi)*(A(n,1) -A(n-1,1)) ...
11                 + sin(th)*sin(phi)*(A(n,2) -A(n-1,2)) + cos(th)*(A(n,3)-A(n-1,3))) );
12     I(n) = exp(j*Psi(n));
13 end
14 phase = Psi;
15 end

```

A.5.3 excitationVector1.m

```

1 function [ I , phase] = excitationVector1( th, phi , A)
2 %UNTITLED4 Summary of this function goes here
3 % Detailed explanation goes here
4 I = zeros(length(A(:,1)),1);
5 Psi = zeros(length(A(:,1)),1);

```

```

6
7 for n = 1:length(A(:,1))
8     Psi(n) = (2*pi *abs(sin(th)*cos(phi)*A(n,1) ...
9             + sin(th)*sin(phi)*A(n,2) + cos(th)*A(n,3) ));
10     I(n) = exp(j*Psi(n));
11 end
12 phase = Psi;
13 end

```

A.6 Task 6

A.6.1 Task6_a.m

```

1 %% part a: H plane --> phi = 0;
2 % task 3, prog 2: is the best program so far
3 clear; clc; close all;
4
5 I =1; % set the current
6
7 % define theta part of G
8 G_th = @(th, phi) 12*pi*abs(I) *(cos(th)*sin(phi))*sin(pi/2*sin(th));
9
10 % define phi part of G
11 G_phi = @(th,phi) 12*pi*abs(I) *(cos(phi))*sin(pi/2*sin(th));
12
13 % define total G
14 G = @(th, phi) [G_th(th, phi) G_phi(th,phi)];
15
16 % task 6:
17 PSI = @(th, thMax) 4*pi/3 *(cos(th)- cos(thMax));
18 AFabs = @(th, thMax, N) 1/N * sin(2*N*pi/3* PSI(th, thMax))...
19         *1/(sin(2*pi/3* PSI(th, thMax)));
20 %% Assume dipole in y direction
21 % define parameters
22 N=5;
23 theta = linspace(0,pi);
24 P= totalPwr(G);
25
26 % element radiation pattern
27 F = @(th, phi) sqrt(1 - sin(th)^2*cos(phi)^2);
28
29 % tot radiation pattern = F * AF
30 D = @(th, phi, thMax, P, N) 10*log(4*pi/P *norm(F(th, phi)*AFabs(th, thMax,N))^2);
31 GD = @(th, phi, thMax, P, N) 10*log(4*pi/sqrt(P) *norm(F(th, phi)*AFabs(th, thMax,N)));
32
33 % plot

```

```

34 for n = 1:length(theta)
35     Dir(n) = (D(theta(n),0, 0, P, N ));
36     GDir(n) = (GD(theta(n),0, 0, P, N ));
37     ArrPat(n) = AFabs(theta(n), 0, N);
38 end
39 figure(1)
40 plot(theta, Dir)
41 xlabel( Angle $\theta$ [rad] , Interpreter , latex , Fontsize ,20)
42 ylabel( Radiation Pattern [dBi] , Interpreter , latex , Fontsize ,20)
43 title( Radiation Pattern in H-plane for y polarization , Interpreter , latex , Fontsize ,20)
44
45 figure(2)
46 plot(theta, GDir)
47 xlabel( Angle $\theta$ [rad] , Interpreter , latex , Fontsize ,20)
48 ylabel( Far field Pattern [dBi] , Interpreter , latex , Fontsize ,20)
49 title( Far field Pattern in H-plane for y polarization , Interpreter , latex , Fontsize ,20)
50
51 figure(3)
52
53     plot(theta, ArrPat)
54 xlabel( Angle $\theta$ [rad] , Interpreter , latex , Fontsize ,20)
55 ylabel( Array factor Pattern , Interpreter , latex , Fontsize ,20)
56
57
58 %% Assume dipole in z direction
59 N = 5;
60 theta = linspace(0,pi);
61 P= totalPwr(G);
62
63 % element radiation pattern
64 F = @(th, phi) sin(th);
65
66 % tot radiation pattern = F * AF
67 D = @(th, phi, thMax, P, N) log(4*pi/P *norm(F(th, phi)*AFabs(th, thMax,N))^2);
68
69 % plot
70 for n = 1:length(theta)
71     Dir(n) = abs(D(theta(n),0, 0, P, N ));
72 end
73 figure(2)
74 plot(theta, Dir)
75
76 xlabel( Angle $\theta$ [rad] , Interpreter , latex , Fontsize ,20)
77 ylabel( Radiation Pattern [dBi] , Interpreter , latex , Fontsize ,20)
78 title( Radiation Pattern in H-plane for z polarization , Interpreter , latex , Fontsize ,20)

```

A.6.2 Task6 b.m

```

1 %% part a: H plane --> phi = 0;

```

```

2 % task 3, prog 2: is the best program so far
3 clear; clc; clf;
4
5 I =1; % set the current
6
7 % define theta part of G
8 G_th = @(th, phi) 12*pi*abs(I) *(cos(th)*sin(phi))*sin(pi/2*sin(th));
9
10 % define phi part of G
11 G_phi = @(th,phi) 12*pi*abs(I) *(cos(phi))*sin(pi/2*sin(th));
12
13 % define total G
14 G = @(th, phi) [G_th(th, phi) G_phi(th,phi)];
15
16 % task 6:
17 PSI = @(th, thMax, lnr) 2*pi/lnr *(cos(th)- cos(thMax));
18 AFabs = @(th, thMax, N, lnr) 1/N * sin(N*pi/lnr* PSI(th, thMax, lnr))...
19         *1/(sin(pi/lnr* PSI(th, thMax, lnr)));
20 %% Assume dipole in y direction
21 vec = [0.25 0.75 1.5 1.5 3.5 4]
22 for pic = 1:6
23     lnr = vec(pic)
24     % define parameters
25     N=5;
26     theta = linspace(0,pi);
27     P= totalPwr(G);
28
29     % element radiation pattern
30     F = @(th, phi) sqrt(1 - sin(th)^2*cos(phi)^2);
31
32     % tot radiation pattern = F * AF
33     D = @(th, phi, thMax, P, N, lnr) 10*log(4*pi/P *norm(F(th, phi)*AFabs(th, thMax,N, lnr))^2);
34     GD = @(th, phi, thMax, P, N, lnr) 10*log(4*pi/sqrt(P) *norm(F(th, phi)*AFabs(th, thMax,N, lnr)));
35
36     % plot
37     for n = 1:length(theta)
38         Dir(n) = (D(theta(n),pi/2, pi/2, P, N , lnr));
39         GDdir(n) = (GD(theta(n),pi/2, pi/2, P, N , lnr))
40         ArrPat(n) = AFabs(theta(n), pi/2, N, lnr);
41     end
42     figure(1)
43     subplot(3,2,pic)
44     plot(theta, Dir)
45
46 xlabel([ Angle $\theta$ [rad], d$\lambda$ = num2str(vec(pic))], Interpreter , latex , ←
47         Fontsize ,20)
48 ylabel( Radiation Pattern [dBi] , Interpreter , latex , Fontsize ,20)
49
50 figure(2)
51 subplot(3,2,pic)
52 plot(theta, GDdir)

```



```

53 xlabel([ Angle $\theta$ [rad], d$\lambda$ = num2str(vec(pic))], Interpreter , latex , ↵
    Fontsize ,20)
54 ylabel( Far field Pattern [dBi] , Interpreter , latex , Fontsize ,20)
55
56
57 figure(3)
58     subplot(3,2,pic)
59     plot(theta, ArrPat)
60 xlabel([ Angle $\theta$ [rad], d$\lambda$ = num2str(vec(pic))], Interpreter , latex , ↵
    Fontsize ,20)
61 ylabel( Array factor Pattern , Interpreter , latex , Fontsize ,20)
62 end
63
64 %% Assume dipole in z direction
65 vec = [0.25 0.75 1.5 1.5 3.5 4]
66 for pic = 1:6
67     lnr = vec(pic)
68     N = 5;
69     theta = linspace(0,pi);
70     P= totalPwr(G);
71
72     % element radiation pattern
73     F = @(th, phi) sin(th);
74
75     % tot radiation pattern = F * AF
76     D = @(th, phi, thMax, P, N, lnr) log(4*pi/P *norm(F(th, phi)*AFabs(th, thMax,N, lnr))^2);
77
78     % plot
79     for n = 1:length(theta)
80         Dir(n) = abs(D(theta(n),pi/2, pi/2, P, N, lnr ));
81         ArrPat(n) = AFabs(theta(n), pi/2, N, lnr);
82     end
83     figure(2)
84     subplot(3,2,pic)
85     plot(theta, Dir)
86 xlabel([ Angle $\theta$ [rad], d$\lambda$ = num2str(vec(pic))], Interpreter , latex , ↵
    Fontsize ,20)
87 ylabel( Radiation Pattern [dBi] , Interpreter , latex , Fontsize ,20)
88
89 figure(3)
90     subplot(3,2,pic)
91     plot(theta, ArrPat)
92 xlabel([ Angle $\theta$ [rad], d$\lambda$ = num2str(vec(pic))], Interpreter , latex , ↵
    Fontsize ,20)
93 ylabel( Array factor Pattern [dBi] , Interpreter , latex , Fontsize ,20)
94 end

```

A.6.3 totalPwr.m

```

1 function [ Prad ] = totalPwr(G)
2 %UNTITLED15 Summary of this function goes here
3 % Detailed explanation goes here
4 % G = [Gth(th, phi) Gphi(th, phi)] is a function
5
6 % Polarization:
7 e=pi/2;
8
9 % vectors for integration
10 phi = linspace(0,2*pi);
11 th = linspace(0,pi);
12
13 for n=1:length(phi)
14     costar = [conj(cos(phi(n)-e))
15              conj(-sin(phi(n)-e))];
16     xpstar = [conj(-sin(phi(n)-e))
17              conj(-cos(phi(n)-e))];
18     for m = 1:length(th)
19         Gco = G(th(m),phi(n))*costar;
20         Gxp = G(th(m),phi(n))*xpstar;
21         U(n,m) = norm(Gco)^2 + norm(Gxp)^2;
22     end
23 end
24
25
26 Prad =0;
27 h= (2*pi^2/(length(phi)*length(th)));
28 % integration with trapez method.
29 for n = 1:length(th)
30     for m = 1:length(phi)
31         Prad = Prad + U(n,m)*sin(th(n))*h;
32     end
33 end
34
35 end

```

Bibliography

- [1] P. Kildal, *Foundations of Antennas: A Unified Approach*. Professional Publishing Svc., 2000. [Online]. Available: <https://books.google.se/books?id=aVOAPQAACAAJ>
- [2] M. W. Blake, Lamont V. Long, *Antennas - Fundamentals, Design, Measurement (3rd Edition)*. SciTech Publishing, 2009. [Online]. Available: <http://app.knovel.com/hotlink/toc/id:kpAFDME001/antennas-fundamentals/antennas-fundamentals>