

Scilab Textbook Companion for  
Antenna and Wave Propagation  
by G. S. N. Raju<sup>1</sup>

Created by  
Abhishek Sharma  
B.E.  
Electronics Engineering  
Madhav Institute of Technology and Science, Gwalior  
College Teacher  
None  
Cross-Checked by  
Harpreeth Singh

June 3, 2016

<sup>1</sup>Funded by a grant from the National Mission on Education through ICT, <http://spoken-tutorial.org/NMEICT-Intro>. This Textbook Companion and Scilab codes written in it can be downloaded from the "Textbook Companion Project" section at the website <http://scilab.in>

# Book Description

**Title:** Antenna and Wave Propagation

**Author:** G. S. N. Raju

**Publisher:** Dorling Kindersley, India

**Edition:** 1

**Year:** 2008

**ISBN:** 9788131701843

Scilab numbering policy used in this document and the relation to the above book.

**Exa** Example (Solved example)

**Eqn** Equation (Particular equation of the above book)

**AP** Appendix to Example(Scilab Code that is an Appednix to a particular Example of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means a scilab code whose theory is explained in Section 2.3 of the book.

# Contents

List of Scilab Codes	4
1 MATHEMATICAL PRELIMINARIES	5
2 MAXWELL EQUATIONS AND ELECTROMAGNETIC WAVES	15
3 RADIATION AND ANTENNAS	34
4 ANALYSIS OF LINEAR ARRAYS	37
6 HF VHF AND UHF ANTEENAS	48
7 MICROWAVE ANTENNAS	61
9 WAVE PROPAGATION	75

# List of Scilab Codes

Exa 1.1	magnitude and direction of a vector . . . . .	5
Exa 1.2	Addition and Subtraction of two vectors . . . . .	5
Exa 1.3	Dot product of two vectors . . . . .	6
Exa 1.4	cross product of two vectors A and B . . . . .	6
Exa 1.5	Dot product of two vectors . . . . .	7
Exa 1.9	Representation of point in cylindrical coordinates . . . .	7
Exa 1.10	Representation of point in cylindrical coordinates . . . .	8
Exa 1.12	Representation of point in spherical coordinates . . . . .	8
Exa 1.17	Power gain in Decibels . . . . .	9
Exa 1.18	Current gain in Decibels . . . . .	9
Exa 1.19	Power gain in nepers . . . . .	10
Exa 1.20	Current gain in nepers . . . . .	10
Exa 1.21	magnitude and phase of a complex number . . . . .	10
Exa 1.22	magnitude complex conjugate and phase of complex number . . . . .	11
Exa 1.23	real and imaginary part of a complex number . . . . .	11
Exa 1.24	Addition of two complex numbers . . . . .	12
Exa 1.25	Subtraction of two complex numbers . . . . .	12
Exa 1.26	Product of two complex numbers . . . . .	13
Exa 1.27	ratio of two complex numbers . . . . .	13
Exa 1.28	Roots of the quadratic equation . . . . .	13
Exa 1.31	factorial of 4 and 6 . . . . .	14
Exa 2.8	magnetic field and its magnitude . . . . .	15
Exa 2.9	electric field and electric flux density . . . . .	16
Exa 2.12	frequency wavelength intrinsic impedance and phase constant . . . . .	17
Exa 2.14	propagation constant . . . . .	18
Exa 2.15	amplitude frequency wavelength and phase constant . . . . .	18

Exa 2.16	electric field in free space and in medium . . . . .	19
Exa 2.17	propagation constant and intrinsic impedance . . . . .	19
Exa 2.18	frequency and permittivity . . . . .	20
Exa 2.19	frequency phase constant and wavelength . . . . .	21
Exa 2.20	conducting characteristics of earth . . . . .	22
Exa 2.21	attenuation constant phase constant phase velocity prop- agation constant and intrinsic impedance . . . . .	23
Exa 2.22	depth of penetration . . . . .	24
Exa 2.23	displacement current . . . . .	25
Exa 2.25	reachable depth of the sea . . . . .	26
Exa 2.27	Power per unit area . . . . .	28
Exa 2.28	average power and maximum energy density of wave . . . . .	28
Exa 2.29	energy density and total energy . . . . .	29
Exa 2.30	transmitted distance of an electromagnetic wave . . . . .	30
Exa 2.31	incident and reflected magnetic field and reflected elec- tric field . . . . .	31
Exa 2.32	average power density absorbed . . . . .	32
Exa 3.1	radiation resistance . . . . .	34
Exa 3.3	Directivity of half wave dipole . . . . .	35
Exa 3.4	Power radiated . . . . .	35
Exa 3.5	effective area of half wave dipole . . . . .	36
Exa 3.6	effective area of hertzian dipole . . . . .	36
Exa 4.1	null to null beam width of a broadside array . . . . .	37
Exa 4.2	null to null beam width of a endfire array . . . . .	38
Exa 4.3	null to null beam width and directivity . . . . .	39
Exa 4.4	Progressive phase shift and array length . . . . .	39
Exa 4.5	null to null and half power beam width and directivity . . . . .	40
Exa 4.9	relative excitation levels . . . . .	41
Exa 4.10	basic and actual transmission loss . . . . .	41
Exa 4.11	basic transmission loss . . . . .	42
Exa 4.12	Actual transmission loss . . . . .	44
Exa 4.13	receiving power . . . . .	46
Exa 6.1	Designing of a rhombic antenna . . . . .	48
Exa 6.2	Designing of a rhombic antenna . . . . .	49
Exa 6.3	Designing of a rhombic antenna . . . . .	49
Exa 6.4	Design parameters of rhombic antenna . . . . .	51
Exa 6.5	Design parameters of rhombic antenna . . . . .	52
Exa 6.6	Design a three element yagi uda antenna . . . . .	53

Exa 6.7	Designing of a six element yagi uda antenna . . . . .	53
Exa 6.8	Designing of a long periodic antenna . . . . .	54
Exa 6.9	induced voltage in a loop antenna . . . . .	56
Exa 6.10	radiation resistance of a loop antenna . . . . .	57
Exa 6.11	Directivity of a loop antenna . . . . .	57
Exa 6.13	Directivity and radiation resistance of a loop antenna .	58
Exa 6.14	array length number of elements and null to null beam width . . . . .	59
Exa 6.15	null to null and half power beam width and directivity	59
Exa 7.1	null to null and half power beam width of a paraboloid reflector . . . . .	61
Exa 7.2	gain of the paraboloid reflector antenna . . . . .	61
Exa 7.3	band width between first null and half power points .	62
Exa 7.4	Power gain in Decibels . . . . .	63
Exa 7.5	mouth diameter HPBW and power gain . . . . .	63
Exa 7.6	beam width directivity and capoture area . . . . .	64
Exa 7.7	minimum distance required between two antennas . .	65
Exa 7.8	mouth diameter and beam width . . . . .	65
Exa 7.9	capture area and beam width of paraboloid antenna .	66
Exa 7.10	HPBW BWFN and power gain . . . . .	67
Exa 7.11	power gain . . . . .	67
Exa 7.12	mouth diameter and capture area of a paraboloid antenna	68
Exa 7.13	mouth diameter and power gain of paraboloid reflector antenna . . . . .	69
Exa 7.14	null to null beam width and power gain of paraboloid reflector antenna . . . . .	69
Exa 7.15	Power gain of a paraboloid reflector antenna . . . . .	70
Exa 7.16	null to null and half power beam width and directivity	70
Exa 7.17	beam width power gain and directivity . . . . .	71
Exa 7.18	power gain of a square horn antenna . . . . .	72
Exa 7.19	power gain and directivity of a horn antenna . . . . .	72
Exa 7.20	complementary slot impedances . . . . .	73
Exa 9.1	Required transmitter power . . . . .	75
Exa 9.2	field strength of the ground wave . . . . .	76
Exa 9.3	maximum range of tropospheric transmission . . . . .	81
Exa 9.4	Radio horizon distance . . . . .	81
Exa 9.5	maximum distance covered by the space wave . . . . .	82
Exa 9.6	Required height of receiving antenna . . . . .	82

Exa 9.7	Radio horizon distance for transmitting and receiving antenna and maximum range . . . . .	83
Exa 9.8	range of the space wave . . . . .	83
Exa 9.9	maximum wavelength at which propagation is possible	84
Exa 9.10	Electron density of the layer . . . . .	85
Exa 9.11	range if the frequency is MUF itself . . . . .	85
Exa 9.12	Relative permittivity of D E F layers . . . . .	86
Exa 9.13	Angle of refraction . . . . .	87
Exa 9.14	critical frequency of an electromagnetic wave . . . . .	87
Exa 9.15	Critical frequency . . . . .	88
Exa 9.16	maximum usable frequency . . . . .	89
Exa 9.17	frequency of propagating wave for D layer . . . . .	89
Exa 9.18	Range of line of sight . . . . .	90
Exa 9.19	Critical angle of propagation for D layer . . . . .	90



# Chapter 1

## MATHEMATICAL PRELIMINARIES

Scilab code Exa 1.1 magnitude and direction of a vector

```
1 //Exa 1.1
2 clc;
3 clear;
4 close;
5 // given :
6 A=[1 2 3] // A is a vector
7 l=norm(A) // magnitude or length of vector A
8 a=A/norm(A) // direction of vector A
9 disp(l,"magnitude of vector")
10 disp(a,"direction of vector")
```

---

Scilab code Exa 1.2 Addition and Subtraction of two vectors

```
1 //Exa 1.2
2 clc;
3 clear;
```

```

4 close;
5 // given :
6 A=[2 5 6] // vector A
7 B=[1 -3 6] // vector B
8 A+B // summation of two vectors
9 A-B // subtraction of two vectors
10 disp(A+B,"summation of two vectors:")
11 disp(A-B,"subtraction of two vectors:")

```

---

### Scilab code Exa 1.3 Dot product of two vectors

```

1 // Exa 1.3
2 clc;
3 clear;
4 close;
5 // given :
6 A=[1 1 2] // vector A
7 B=[2 1 1] // vector B
8 k=sum(A.*B) // dot product of vector A and B
9 disp(k,"dot product of vector A and B:")

```

---

### Scilab code Exa 1.4 cross product of two vectors A and B

```

1 //Exa 1.4
2 clc;
3 clear;
4 close;
5 function [V]=crossprod(A,B) // defining a function v
6 V(1)=A(2)*B(3)-A(3)*B(2)
7 V(2)=A(3)*B(1)-A(1)*B(3)
8 V(3)=A(1)*B(2)-A(2)*B(1)
9 endfunction
10 //given:

```

```
11 A=[2,1,2] // vector A
12 B=[1,2,1] // vector B
13 P=crossprod(A,B)
14 disp(P,"cross product of vectors A and B:")
```

---

#### Scilab code Exa 1.5 Dot product of two vectors

```
1 // Exa 1.5
2 clc;
3 clear;
4 close;
5 // given :
6 A=[1 3 4] // vector A
7 B=[1 0 2] // vector B
8 k=sum(A.*B) // dot product of two vectors A and B
9 disp(k,"dot product of two vectors A and B:")
```

---

#### Scilab code Exa 1.9 Representation of point in cylindrical coordinates

```
1 //Exa 1.9
2 clc;
3 clear;
4 close;
5 // given :
6 p=[1,2,3] // coordinates of point p
7 x=1 // x coordinate of P
8 y=2 // y coordinate of P
9 z=3 // z coordinate of P
10 rho=sqrt(x^2+y^2) //radius of cylinder in m
11 phi=atand(y/x) // azimuthal angle in degrees
12 z=3 // in m
13 disp(rho,"radius of cylinder in m:")
14 disp(phi,"azimuthal angle in degrees:")
```

```
15 disp(z,"z coordinate in Degrees:")
```

---

**Scilab code Exa 1.10** Representation of point in cylindrical coordinates

```
1 //Exa 1.10
2 clc;
3 clear;
4 close;
5 // given :
6 A=[4,2,1] // vector A
7 A_x=4 // x coordinate of P
8 A_y=2 // y coordinate of P
9 A_z=1 // z coordinate of P
10 phi=atand(A_y/A_x) // azimuthal in degrees
11 A_rho=A_x*cosd(phi)+A_y*sind(phi) // x coordinate of
    cylinder
12 A_phi=-A_x*sind(phi)+A_y*cosd(phi) // y coordinate
    of cylinder
13 A_z=1 // z coordinate of cylinder
14 A=[A_rho,A_phi,A_z] // cylindrical coordinates if
    vector A
15 disp(A,"cylindrical coordinates of vector A:")
```

---

**Scilab code Exa 1.12** Representation of point in spherical coordinates

```
1 // Exa 1.12
2 clc;
3 clear;
4 close;
5 // given :
6 P=[1,2,3] // coordinates of point P in cartezian
    system
7 x=1 // x coordinate of point P in cartezian system
```

```

8 y=2// y coordinate of point P in cartezian system
9 z=3// z coordinate of point P in cartezian system
10 r=sqrt(x^2+y^2+z^2) // radius of sphere in m
11 theta=acosd(z/(r)) // angle of elevation in degrees
12 phi=atand(x/y) // azimuthal angle in degrees
13 disp(r,"radius of sphere in m:")
14 disp(theta,"angle of elevation in degrees:")
15 disp(phi,"azimuthal angle in degrees:")
16
17
18 // note : answer in the book is incomplete they find
    only one coordinate but there are three.

```

---

#### Scilab code Exa 1.17 Power gain in Decibels

```

1 // Exa 1.17
2 clc;
3 clear;
4 close;
5 // given :
6 A_p=22 // power gain
7 A_p_dB=10*log10(A_p) // power gain in dB
8 disp(A_p_dB,"power gain in dB:")

```

---

#### Scilab code Exa 1.18 Current gain in Decibels

```

1 // Exa 1.18
2 clc;
3 clear;
4 close;
5 // given :
6 A_v=95 // voltage gain
7 A_v_dB=20*log10(A_v) // voltage gain in dB

```

```
8 disp(A_v_dB,"voltage gain in dB:")
```

---

**Scilab code Exa 1.19** Power gain in nepers

```
1 // Exa 1.19
2 clc;
3 clear;
4 close;
5 // given :
6 A_p=16 // power gain
7 A_p_Np=log(sqrt(A_p)) // power gain in nepers
8 disp(A_p_Np,"power gain in nepers:")
```

---

**Scilab code Exa 1.20** Current gain in nepers

```
1 // Exa 1.20
2 clc;
3 clear;
4 close;
5 // given :
6 A_i=34 // current gain
7 A_i_Np=log(A_i) // current gain in nepers
8 disp(A_i_Np,"current gain in nepers:")
```

---

**Scilab code Exa 1.21** magnitude and phase of a complex number

```
1 // Exa 1.21
2 clc;
3 clear;
4 close;
```

```

5 // given :
6 A=2+4*i // complex number A
7 magnitude=sqrt((real(A))^2+(imag(A))^2) // magnitude
  of complex number A
8 phi=atand(imag(A)/real(A)) // phase of complex
  number A in degrees
9 disp(magnitude,"magnitude of complex number A:")
10 disp(phi,"phase of complex number A in degrees:")

```

---

**Scilab code Exa 1.22** magnitude complex conjugate and phase of complex number

```

1 // Exa 1.22
2 clc;
3 clear;
4 close;
5 // given :
6 A=1+3*i // complex no. A
7 c=conj(A) // conjugate of complex no. A
8 magnitude=sqrt((real(A))^2+(imag(A))^2) // magnitude
  of complex number A
9 phi=atand(imag(A)/real(A)) // phase of complex
  number A in degrees
10 disp(magnitude,"magnitude of complex number A:")
11 disp(phi,"phase of complex number A in degrees:")
12 disp(c,"conjugate of complex no. A:")

```

---

**Scilab code Exa 1.23** real and imaginary part of a complex number

```

1 // Exa 1.23
2 clc;
3 clear;
4 close;

```

```

5 // given :
6 rho=5 // magnitude of the complex number A
7 phi=45 // phase of a complex number A in Degrees
8 x=rho*cosd(phi) // real part of complex number A
9 y=rho*sind(phi) // imaginary part of complex number
  A
10 A=x+y*(%i) // complex number A
11 disp(x,"real part of complex number A:")
12 disp(y,"imaginary part of complex number A:")
13 disp(A,"complex number A:")

```

---

**Scilab code Exa 1.24** Addition of two complex numbers

```

1 // Exa 1.24
2 clc;
3 clear;
4 close;
5 // given :
6 A_1=2+%i*3 // complex number A_1
7 A_2=4+%i*5 // complex number A_2
8 A=A_1+A_2
9 disp(A,"sum of complex numbers A_1 and A_2:")

```

---

**Scilab code Exa 1.25** Subtraction of two complex numbers

```

1 // Exa 1.25
2 clc;
3 clear;
4 close;
5 // given :
6 A_1=%i*6 // complex number A_1
7 A_2=1-%i*2 // complex number A_2
8 A=A_1-A_2

```



```
9 disp(A," difference of complex numbers A_1 and A_2:")
```

---

**Scilab code Exa 1.26** Product of two complex numbers

```
1 // Exa 1.26
2 clc;
3 clear;
4 close;
5 // given :
6 A=0.4+%i*5 // complex number A
7 B=2+%i*3 // complex number B
8 P=A*B // product of complex numbers A and B
9 disp(P,"product of complex numbers A and B:")
```

---

**Scilab code Exa 1.27** ratio of two complex numbers

```
1 // Exa 1.27
2 clc;
3 clear;
4 close;
5 // given :
6 A=10+%i*6 // complex number A
7 B=2-%i*3 // complex number B
8 D=A/B // division of complex numbers A and B
9 disp(D,"division of complex numbers A and B:")
```

---

**Scilab code Exa 1.28** Roots of the quadratic equation

```
1 //Exa 1.28
2 clc;
```

```
3 clear;
4 close;
5 // given :
6 x=poly([0], 'x')
7 p=(x)^2+2*x+4
8 roots(p) // roots of given quadratic equation
9 disp(roots(p),"The roots of the given quadratic
    equation are:")
```

---

**Scilab code Exa 1.31** factorial of 4 and 6

```
1 //Exa 1.31
2 clc;
3 clear;
4 close;
5 f1=factorial(4) // factorial of 4
6 f2=factorial(6) // factorial of 6
7 disp(f1,"factorial of 4 is:")
8 disp(f2,"factorial of 6 is:")
```

---

## Chapter 2

# MAXWELL EQUATIONS AND ELECTROMAGNETIC WAVES

Scilab code Exa 2.8 magnetic field and its magnitude

```
1 // Exa 2.8
2 clc;
3 clear;
4 close;
5 // given :
6 mu_0=4*%pi*10^(-7) // permeability in free space
7 mu_r1=3 // region 1 relative permeability
8 mu_r2=5 // region 2 relative permeability
9 mu_1=mu_r1*mu_0 // region 1 permeability
10 mu_2=mu_r2*mu_0 // region 2 permeability
11 H1=[4 1.5 -3] // magnetic field in region 1 in A/m
12 Ht1=[0 1.5 -3] // tangential component of magnetic
    field H1
13 Hn1=[4 0 0] // normal component of magnetic field H1
14 Ht2=[0 1.5 -3] // as tangential component of
    magnetic field H2=tangential component of
    magnetic field H1
```

```

15 Hn2=(mu_1/mu_2)*Hn1 // normal component of magnetic
    field H2
16 H2=Ht2+Hn2 // magnetic field in region 2 in A/m
17 h2=norm(H2) // magnitude of the magnetic field H2 in
    A/m
18 disp(H2,"magnetic field in region 2 in A/m:")
19 disp(h2,"magnitude of magnetic field in region 2 in
    A/m:")

```

---

### Scilab code Exa 2.9 electric field and electric flux density

```

1 // Exa 2.9
2 clc;
3 clear;
4 close;
5 // given :
6 epsilon_0=8.854*10^(-12) // permittivity in free
    space
7 sigma_1=0 //conductivity of medium 1
8 sigma_2=0 // conductivity of medium 2
9 epsilon_r1=1 // region 1 relative permittivity
10 epsilon_r2=2 // region 2 relative permittivity
11 epsilon_1=epsilon_r1*epsilon_0 // region 1
    permittivity
12 epsilon_2=epsilon_r2*epsilon_0 // region 2
    permittivity
13 E1=[1 2 3] // Electric field in region 1 in V/m
14 Et1=[0 2 3] // tangential component of electric
    field E1
15 En1=[1 0 0] // normal component of electric field E1
16 Et2=[0 2 3] // as tangential componenet of electric
    field E2=tangential component of electric field
    E1
17 En2=(epsilon_1/epsilon_2)*En1 // normal component of
    electric field E2

```

```

18 E2=Et2+En2 // electric field in region 2 in V/m
19 Dt1=epsilon_0*Et1 // tangential component of
    electric flux density D1
20 D2=epsilon_2*E2 // electric flux density in region 2
    in C/m^2
21 disp(E2,"electric field in region 2 in V/m:")
22 disp(D2,"electric flux density in region 2 in C/m^2:
    ")

```

---

**Scilab code Exa 2.12** frequency wavelength intrinsic impedance and phase constant

```

1 // Exa 2.12
2 clc;
3 clear;
4 close;
5 // given :
6 //H=cos(10^8*t-Beta*z)ay // magnetic field in A/m
7 // E=377*cos(10^8*t-Beta*z)ax // electric field in
    V/m
8 omega=10^8 // angular frequency in Hz
9 f=omega/(2*pi) // frequency in Hz
10 v_0=3*10^8 // speed of light in m/s
11 lambda=v_0/f // wavelength in m
12 Beta=2*pi/lambda // phase constant in rad/m
13 disp("eta_0= E/H = 377*cos(10^8*t-Beta*z)/cos(10^8*t
    -Beta*z) => E/H=377")
14 eta_0=abs(377) // intrinsic impedance in ohm
15 disp(eta_0,"intrinsic impedance in ohm:")
16 disp(f/10^6,"frequency in MHz:")
17 disp(Beta,"phase constant in rad/m:")
18 disp(lambda,"wavelength in m:")
19
20 //note : answer of lambda in book is rounded-off, it
    is 18.86 meter.

```

---

**Scilab code Exa 2.14** propagation constant

```
1 // Exa 2.14
2 clc;
3 clear;
4 close;
5 // given :
6 f=100 // frequency in MHz
7 f=100*10^6 // frequency in Hz
8 v_0=3*10^8 // speed of light in m/s
9 // formula :  $\Gamma = i \omega \sqrt{\mu_0 \epsilon_0} = i \omega / v_0 = (i 2 \pi f) / v_0$ 
10 Gamma=i*2*pi*f/v_0 //propagation constant
11 disp(Gamma,"propagation constant in m-1:")
12 funcprot(0)
```

---

**Scilab code Exa 2.15** amplitude frequency wavelength and phase constant

```
1 // Exa 2.15
2 clc;
3 clear;
4 close;
5 // given :  $H(z,t) = 48 \cos(10^8 t + 40 z) \mathbf{a}_y$  // equation
   // of magnetic field
6 A=48 // amplitude of the magnetic field in A/m
7 omega=10^8 // angular frequency in radians/sec
8 f=omega/(2*pi) // frequency in Hz
9 Beta=40 // phase constant in rad/m
10 lambda=2*pi/Beta // wavelength in m
11 disp(A,"amplitude of the magnetic field in A/m:")
12 disp(f/10^6,"frequency in MHz:")
```

```
13 disp(Beta,"phase constant in rad/m:")
14 disp(lambda,"wavelength in m:")
```

---

**Scilab code Exa 2.16** electric field in free space and in medium

```
1 // Exa 2.16
2 clc;
3 clear;
4 close;
5 // given :
6 H=2 // amplitude of magnetic field in A/m
7 sigma=0 // conductivity
8 mu_0=4*pi*10^-7 // permeability in free space in H/
   m
9 epsilon_0=8.854*10^-12 // permittivity in free space
   in F/m
10 mu=mu_0 // permeability in F/m
11 epsilon=4*epsilon_0 // permittivity in F/m
12 Eta_0=120*pi // intrinsic impedance in free space
   in ohm
13 E=Eta_0*H // electric field in V/m
14 disp(E,"magnitude of electric field in V/m in free
   space:")
15 Eta=sqrt(mu/epsilon) // intrinsic impedance in ohm
16 E=Eta*H // magnitude of electric field
17 disp(E,"magnitude of electric field in V/m:")
```

---

**Scilab code Exa 2.17** propagation constant and intrinsic impedance

```
1 //Exa 2.17
2 clc;
3 clear;
4 close;
```

```

5 // given :
6 sigma=0 // conductivity in mho/m
7 f=0.3 // frequency in GHz
8 f=0.3*10^9 // frequency in Hz
9 omega=2*pi*f // angular frequency in rad/sec
10 // formula : Gamma=sqrt(%i*omega*mu*(sigma+%i*omega*
    epsilon))=%i*omega*sqrt(mu*epsilon)
11 epsilon_0=8.854*10^-12 // permittivity in free space
    in F/m
12 epsilon=9*epsilon_0 // permittivity in F/m
13 mu_0=4*pi*10^-7 // permeability in free space in H
    /m
14 mu=mu_0 // permeability in H/m
15 Gamma=%i*omega*sqrt(mu*epsilon) // propagation
    constant in m^-1
16 disp(Gamma,"propagation constant in m^-1:")
17 // formula : eta=sqrt((%i*omega*mu)/(sigma+omega*
    epsilon))=sqrt(mu/epsilon)
18 eta=sqrt(mu_0/(9*epsilon_0)) // intrinsic impedance
    in ohm
19 disp(eta,"intrinsic impedance in ohm:")
20
21
22 // note : answer of propagation constant in book is
    wrong.they put mu_0=4*10^-7 in part 1 which is
    wrong the correct value of mu_0 is 4*pi*10^-7.

```

---

### Scilab code Exa 2.18 frequency and permittivity

```

1 //Exa 2.18
2 clc;
3 clear;
4 close;
5 // given :
6 lambda=0.25 // wavelength in m

```



```

7 v=1.5*10^10 // velocity of propagation of wave in cm
  /sec
8 v=1.5*10^8 // velocity of propagation of wave in m/
  sec
9 f=v/lambda // frequency in Hz
10 disp(f/10^6,"frequency in MHz:")
11 epsilon_0=8.854*10^-12 // permittivity in free space
  in F/m
12 mu_0=4*%pi*10^-7 // permeability in free space in H
  /m
13 mu=mu_0 // permeability in H/m
14 v_0=3*10^8 // speed of light in m/s
15 // formula : v=1/(mu*epsilon)=1/(mu_0*epsilon_0*
  epsilon_r)=v_0/sqrt(epsilon_r)
16 epsilon_r=(v_0/v)^2 // relative permittivity
17 disp(epsilon_r,"relative permittivity:")
18
19
20 //note : answer in the book is wrong.

```

---

### Scilab code Exa 2.19 frequency phase constant and wavelength

```

1 // Exa 2.19
2 clc;
3 clear;
4 close;
5 // given : E=5*sin(10^8*t+4*x)az // equation of
  electric field
6 A=5 // amplitude of the electric field
7 omega=10^8 // angular frequency in radians/sec
8 f=omega/(2*%pi) // frequency in Hz
9 Beta=4 // phase constant in rad/m
10 v_0=3*10^8 // speed of light in m/s
11 lambda=v_0/f // wavelength in m
12 disp(f/10^6,"frequency in MHz:")

```

```
13 disp(Beta,"phase constant in rad/m:")
14 disp(lambda,"wavelength in m:")
```

---

### Scilab code Exa 2.20 conducting characteristics of earth

```
1 // Exa 2.20
2 clc;
3 clear;
4 close;
5 // given :
6 sigma=10^-2 // conductivity of earth in mho/m
7 epsilon_r=10 // relative permittivity
8 mu_r=2 // relative permeability
9 epsilon_0=(1/(36*pi))*10^-9 // permittivity in free
    space
10 epsilon=epsilon_r*epsilon_0 // permittivity
11 f1=50 // frequency in Hz
12 omega1=2*pi*f1 // angular frequency in rad/sec
13 disp("When frequency=50Hz:")
14 k1=sigma/(omega1*epsilon)
15 disp(k1,"K1 is equal to")
16 disp("since k1>>1 hence it behaves like a good
    conductor:")
17 f2=1 // frequency in kHz
18 f2=1*10^3 // frequency in Hz
19 omega2=2*pi*f2 // angular frequency in rad/sec
20 disp("When frequency=1kHz:")
21 k2=sigma/(omega2*epsilon)
22 disp(k2,"K2 is equal to")
23 disp("since k2>>1 hence it behaves like a good
    conductor:")
24 f3=1 // frequency in MHz
25 f3=1*10^6 // frequency in Hz
26 omega3=2*pi*f3 // angular frequency in rad/sec
27 disp("When frequency=1MHz:")
```

```

28 k3=sigma/(omega3*epsilon)
29 disp(k3,"K3 is equal to")
30 disp("since k3=18 hence it behaves like a moderate
      conductor:")
31 f4=100 // frequency in MHz
32 f4=100*10^6 // frequency in Hz
33 omega4=2*pi*f4 // angular frequency in rad/sec
34 disp("When frequency=100MHz:")
35 k4=sigma/(omega4*epsilon)
36 disp(k4,"K4 is equal to")
37 disp("since k4=0.18 hence it behaves like a quasi-
      dielectric:")
38 f5=10 // frequency in GHz
39 f5=10*10^9 // frequency in Hz
40 omega5=2*pi*f5 // angular frequency in rad/sec
41 disp("When frequency=10GHz:")
42 k5=sigma/(omega5*epsilon)
43 disp(k5,"K5 is equal to")
44 disp("since k5<<1 hence it behaves like a good
      dielectric:")

```

---

**Scilab code Exa 2.21** attenuation constant phase constant phase velocity propagation constant and intrinsic impedance

```

1 // Exa 2.21
2 clc;
3 clear;
4 close;
5 // given :
6 f=60 // frequency in Hz
7 omega=2*pi*f // angular frequency in rad/sec
8 sigma=5.8*10^7 // conductivity in mho/m
9 epsilon_0=8.854*10^-12 // permittivity in free space
      in F/m
10 mu_0=4*pi*10^-7 // permeability in free space in H

```

```

    /m
11 epsilon_r=1 // relative permittivity
12 mu_r=1 // relative permeability
13 epsilon=epsilon_r*epsilon_0 // permittivity
14 mu=mu_0*mu_r // permeability
15 k=sigma/(omega*epsilon) // ratio
16 disp(k,"ratio k is equal to")
17 disp("since k>>1 therefore it is very good conductor
    :")
18 alpha=sqrt(omega*mu*sigma/2) // attenuation constant
    in m-1
19 Beta=sqrt(omega*mu*sigma/2) // phase constant in m
    -1
20 Gamma=alpha+(%i*Beta) // propagation constant in m
    -1
21 lambda=2*pi/Beta // wavelength
22 eta=sqrt((%i*omega*mu/sigma)) // intrinsic impedance
    in ohm
23 v=lambda*f // phase velocity of wave in m/s
24 disp(alpha,"attenuation constant in m-1:")
25 disp(Beta,"phase constant in m-1:")
26 disp(Gamma,"propagation constant in m-1:")
27 disp(eta,"intrinsic impedance in ohm:")
28 disp(lambda*100,"wavelength in cm:")
29 disp(v,"phase velocity of wave in m/s:")

```

---

### Scilab code Exa 2.22 depth of penetration

```

1 // Exa 2.22
2 clc;
3 clear;
4 close;
5 // given :
6 f1=60 // frequency in Hz
7 omega1=2*pi*f1 // angular frequency in Hz

```

```

8 f2=100 // frequency in MHz
9 f2=100*10^6 // frequency in Hz
10 omega2=2*pi*f2 // angular frequency in Hz
11 sigma=5.8*10^7 // conductivity in mho/m
12 epsilon_0=8.854*10^-12 // permittivity in free space
    in F/m
13 mu_0=4*pi*10^-7 // permeability in free space in H
    /m
14 epsilon_r=1 // relative permittivity
15 mu_r=1 // relative permeability
16 epsilon=epsilon_r*epsilon_0 // permittivity
17 mu=mu_0*mu_r // permeability
18
19 disp("At f=60Hz")
20 k1=(sigma)/(omega1*epsilon) // ratio
21 disp(k1," ratio k is equal to")
22 disp("since k>>1 therefore it is very good conductor
    at f=60Hz:")
23 delta1=(sqrt(2/(omega1*mu*sigma))) // depth of
    penetration in m
24 disp(delta1,"depth of penetration delta1 in m:")
25
26 disp("At f=100Hz")
27 k2=sigma/(omega2*epsilon) // ratio
28 disp(k2,"ratio k is equal to")
29 disp("since k2>>1 therefore it is very good
    conductor at f=100Hz:")
30 delta2=(sqrt(2/(omega2*mu*sigma))) // depth of
    penetration in m
31 disp(delta2,"depth of penetration delta2 in m:")

```

---

Scilab code Exa 2.23 displacement current

```

1 // Exa 2.23
2 clc;

```

```

3 clear;
4 close;
5 // given :
6 Ic=10 // conduction current in ampere
7 epsilon_r=1 // relative permittivity
8 epsilon_0=8.854*10^-12 // permittivity in free space
9 epsilon=epsilon_r*epsilon_0 // permittivity
10 sigma=5.8*10^7 // conductivity in mho/m
11 disp("when f=1MHz")
12 f=1 // frequency in MHz
13 f=1*10^6 // frequency in Hz
14 Id=2*%pi*f*epsilon*Ic/sigma // displacement current
15 disp(Id,"displacement current when f=1MHz in A:")
16 disp("when f=100MHz")
17 f=100 // frequency in MHz
18 f=100*10^6 // frequency in Hz
19 Id=2*%pi*f*epsilon*Ic/sigma // displacement current
20 disp(Id,"displacement current when f=100MHz in A:")

```

---

Scilab code Exa 2.25 reachable depth of the sea

```

1 // Exa 2.25
2 clc;
3 clear;
4 close;
5 // given :
6 Em=20 // minimum signal level required for vessel
   under sea water in microV/m
7 Em=20*10^-6 // minimum signal level required for
   vessel under sea water in V/m
8 E=100 // electric intensity of wave in V/m
9 v=3*10^8 // speed of light in m/s
10 f=4 // frequency in MHz
11 f=4*10^6 // frequency in Hz
12 omega=2*%pi*f // angular frequency in Hz

```

```

13 sigma=4 // conductivity of sea water in mho/m
14 epsilon_r=81 // relative permittivity
15 epsilon_0=8.854*10^-12 // permittivity in free space
16 epsilon=epsilon_r*epsilon_0 // permittivity
17 mu_r=1 // relative permeability
18 mu_0=4*pi*10^-7 // permeability in free space
19 mu=mu_r*mu_0 // permeability
20 k=(sigma)/(omega*epsilon)//ratio
21 disp("ratio k is equal to: ")
22 disp(k,"ratio:")
23 disp("K is >>1 so sea water is a good conductor")
24 eta_1=377 // intrinsic impedance in free space in
    ohm
25 alpha_1=0 // attenuation constant in free space in m
    ^-1
26 beta_1=omega/v // phase constant in m^-1
27 mageta_2=sqrt(omega*mu/sigma) // magnitude of eta_2(
    intrinsic impedance of sea water in ohm)
28 argeta_2=45 // argument of eta_2 in degrees
29 eta_2=mageta_2*cosd(argeta_2)+%i*mageta_2*sind(
    argeta_2) //intrinsic impedance in complex form (
    r*cos(theta)+%i*r*sin(theta))
30 TC=2*eta_2/(eta_1+eta_2) // transmission coefficient
31 Et=abs(TC)*E // transmitted electric field in V/m
32 alpha_2=sqrt(omega*mu*sigma/2) // attenuation
    constant for sea water in m^-1
33 // formula: Et*exp(-alpha_2*d)=Es
34 d=-(1/alpha_2)*log(Em/Et) // depth in the sea that
    can be reached by the aeroplane in m
35 disp(d,"depth in the sea that can be reached by the
    aeroplane in m:")
36
37
38 // note 1: the value of alpha_2 in book is 7.905 but
    it is "7.94" exactly calculated by scilab
39 //note 2 : The correct answer of the Depth(d) is
    "1.41094" the answer in the book is wrong.

```

**Scilab code Exa 2.27** Power per unit area

```
1 // Exa 2.27
2 clc;
3 clear;
4 close;
5 // given :
6 eta_0=377 // intrinsic impedance in free space in
   ohm
7 disp("E=sin(omega*t-beta*z)ax+2*sin(omega*t-beta*z
   +75)ay // electric field in V/m")
8 Ex=1 // magnitude of Ex
9 Ey=2 // magnitude of Ey
10 E=sqrt(Ex^2+Ey^2) // resultant magnitude
11 Pav=(1/2)*E^2/(eta_0) // power per unit area
   conveyed by the wave in free space
12 disp(Pav*1000,"power per unit area conveyed by the
   wave in free space in mW/m^2:")
```

---

**Scilab code Exa 2.28** average power and maximum energy density of wave

```
1 // Exa 2.28
2 clc;
3 clear;
4 close;
5 // given :
6 epsilon_0=8.854*10^-12 // permittivity in free space
   in F/m
7 mu_0=4*pi*10^-7 // permeability in free space in H
   /m
8 epsilon_r=4 // relative permittivity
9 mu_r=1 // relative permeability
```



```

10 epsilon=epsilon_r*epsilon_0 // permittivity
11 mu=mu_0*mu_r // permeability
12 H=5 // magnitude of magnetic field in mA/m
13 H=5*10^-3 // magnitude of magnetic field in A/m
14 eta=sqrt(mu/epsilon)// intrinsic impedance in ohm
15 E=H*sqrt(mu/epsilon) // magnitude of electric field
16 P_av=E^2/(2*eta) // average power
17 W_E=epsilon*E^2 // maximum energy density of the
    wave
18 disp(P_av*10^6,"Average power in micro*w/m^2:")
19 disp(W_E*10^12,"maximum energy density of the wave
    in PJ/m^3:")
20
21
22 // note: P_av is = 2353.75 in book but it is 2354.58
    correctly calculated by scilab.

```

---

**Scilab code Exa 2.29** energy density and total energy

```

1 // Exa 2.29
2 clc;
3 clear;
4 close;
5 // given :
6 epsilon_0=8.854*10^-12 // permittivity in free space
    in F/m
7 mu_0=4*%pi*10^-7 // permeability in free space in H
    /m
8 epsilon_r=1 // relative permittivity
9 mu_r=1 // relative permeability
10 epsilon=epsilon_r*epsilon_0 // permittivity
11 mu=mu_0*mu_r // permeability
12 E=100*sqrt(%pi) // magnitude of electric field in V/
    m
13 W_E=(1/2)*epsilon*E^2 // electric energy density of

```

```

    the wave
14 disp(W_E*10^9," electric energy density of the wave
    in nJ/m^3:")
15 W_H=W_E // as the energy density is equal to that of
    magnetic field for a plane travelling wave
16 W_T=W_E+W_H // total energy density
17 disp(W_H*10^9," magnetic energy density of wave in nJ
    /m^3:")
18 disp(W_T*10^9," Total energy density in nJ/m^3:")

```

---

**Scilab code Exa 2.30** transmitted distance of an electromagnetic wave

```

1 // Exa 2.30
2 clc;
3 clear;
4 close;
5 // given :
6 sigma=5 // conductivity of sea water in mho/m
7 f1=25 // frequency in kHz
8 f1=25*10^3 // frequency in Hz
9 omega1=2*pi*f1 // angular frequency in Hz
10 f2=25 // frequency in MHz
11 f2=25*10^6 // frequency in Hz
12 omega2=2*pi*f2 // angular frequency in Hz
13 epsilon_r=81 // relative permittivity
14 epsilon_0=8.854*10^-12 // permittivity in free space
15 epsilon=epsilon_r*epsilon_0 // permittivity
16 mu_r=1 // relative permeability
17 mu_0=4*pi*10^(-7) // permeability in free space
18 mu=mu_r*mu_0 // permeability
19 disp("when frequency=25kHz")
20 alpha_1=omega1*sqrt((mu*epsilon)/2*(sqrt(1+(sigma
    ^2/(omega1^2*epsilon^2)))-1)) // attenuation
    constant when f=25kHz
21 // formula: exp(-alpha*x)=0.1

```

```

22 x1=2.3/alpha_1 // transmitted distance in m
23 disp(x1,"transmitted distance in m:")
24 disp("when frequency=25MHz")
25 alpha_2=omega2*sqrt((mu*epsilon)/2*(sqrt(1+(sigma
    ^2/(omega2^2*epsilon^2)))-1)) // attenuation
    constant when f=25MHz
26 x2=2.3/alpha_2 // transmitted distance in m
27 disp(x2,"transmitted distance in m:")
28
29
30 //note: the values of epsilon_r=81 and of mu_r=1 for
    sea water which are not given in the book.

```

---

**Scilab code Exa 2.31** incident and reflected magnetic field and reflected electric field

```

1 // Exa 2.31
2 clc;
3 clear;
4 close;
5 // given :
6 E_i=1 // magnitude of incident electric field in mV/
    m
7 E_i=1*10^-3 // magnitude of incident electric field
    in V/m
8 epsilon_0=8.854*10^-12 // permittivity in free space
    in F/m
9 mu_0=4*%pi*10^-7 // permeability in free space in H
    /m
10 theta_i=15 // incident angle in degrees
11 epsilon_r1=8.5 // relative permittivity of medium 1
12 mu_r1=1 // relative permeability of medium 1
13 epsilon1=epsilon_r1*epsilon_0 // permittivity
14 mu1=mu_0*mu_r1 // permeability
15 eta1=sqrt(mu1/epsilon1) // intrinsic impedance of

```

```

    medium 1 in ohm
16 epsilon2=epsilon_0 // permittivity of medium 2
17 mu2=mu_0 // permeability of medium 2
18 eta2=sqrt(mu2/epsilon2) // intrinsic impedance of
    medium 2 in ohm
19 //formula :  $\sin(\theta_i)/\sin(\theta_t)=\sqrt{\epsilon_2/\epsilon_1}$ 
    epsilon2/epsilon1)
20 theta_t=asind(sind(theta_i)/(sqrt(epsilon2/epsilon1)
    )) // transmitted angle in degrees
21 E_r=E_i*((eta2*cosd(theta_i)-(eta1*cosd(theta_i)))/(
    eta2*cosd(theta_i)+eta1*cosd(theta_i))) //
    reflection coefficient of electric field
22 disp(E_r*1000,"reflection coefficient of electric
    field in mV/m:")
23 H_i=E_i/eta1 // incident coefficient of magnetic
    field
24 disp(H_i*10^6,"incident coefficient of magnetic field
    in micro*A/m:")
25 H_r=E_r/eta1 // reflection coefficient of electric
    field
26 disp(H_r*10^6,"reflection coefficient of magnetic
    field in micro*A/m:")
27
28
29 // note : minute difference in decimal answer
    between scilab and book.

```

---

**Scilab code Exa 2.32** average power density absorbed

```

1 // Exa 2.32
2 clc;
3 clear;
4 close;
5 // given :
6 sigma=5.8*10^7 // conductivity in mho/m

```

```

7 f=2 // frequency in MHz
8 f=2*10^6 // frequency in Hz
9 omega=2*%pi*f // angular frequency in rad/sec
10 E=2 // magnitude of electric field in mV/m
11 E=2*10^-3 // magnitude of electric field in V/m
12 epsilon_0=8.854*10^-12 // permittivity in free space
    in F/m
13 mu_0=4*%pi*10^-7 // permeability in free space in H
    /m
14 epsilon_r=1 // relative permittivity
15 mu_r=1 // relative permeability
16 epsilon=epsilon_r*epsilon_0 // permittivity
17 mu=mu_0*mu_r // permeability
18 eta=sqrt(mu*omega/sigma) // intrinsic impedance in
    ohm
19 P_av=(1/2)*E^2/eta // average power density
    anbsorbed by copper
20 disp(P_av*1000,"average power density anbsorbed by
    copper in mW/m^2:")

```

---

# Chapter 3

## RADIATION AND ANTENNAS

Scilab code Exa 3.1 radiation resistance

```
1 // Exa 3.1
2 clc;
3 clear;
4 close;
5 Lm=poly(0, 'Lm') // defining Lm as lambda
6 dl=Lm/40 // dipole length
7 Rr=80*(%pi)^(2)*(dl/Lm)^2
8 Rr=horner(Rr,1)
9 disp(Rr,"radiation resistance of dipole in ohm if dl
    =Lm/40 :")
10 dl=Lm/60 // dipole length
11 Rr=80*(%pi)^(2)*(dl/Lm)^2
12 Rr=horner(Rr,1)
13 disp(Rr,"radiation resistance of dipole in ohm if dl
    =Lm/60 :")
14 dl=Lm/80 // dipole length
15 Rr=80*(%pi)^(2)*(dl/Lm)^2
16 Rr=horner(Rr,1)
17 disp(Rr,"radiation resistance of dipole in ohm if dl
```

= $I_m/80$  :”)

---

### Scilab code Exa 3.3 Directivity of half wave dipole

```
1 //Exa 3.3
2 clc;
3 clear;
4 close;
5 // given :
6 Pr=1 //power in watt
7 I=sqrt(Pr/73) // current in A
8 Eta0=120*(%pi) // constant
9 r=poly(0, 'r')
10 E_max=60*I/r
11 RI=r^2*E_max^2/Eta0 // radiation intensity
12 Gd_max=4*(%pi)*(RI)/Pr
13 Gd_max=horner(Gd_max,1)
14 disp(Gd_max,"Directivity of a half wave dipole:")
```

---

### Scilab code Exa 3.4 Power radiated

```
1 //Exa3.4
2 clc;
3 clear;
4 close;
5 Rr=300 // radiation resistance in ohm
6 I=3 // in A
7 //formula: Pr=I^2*R
8 Pr=I^2*Rr // power radiated in watt
9 disp(Pr,"power radiated by antenna in watts:")
```

---

**Scilab code Exa 3.5** effective area of half wave dipole

```
1 //Exa 3.5
2 clc;
3 close;
4 clear;
5 // given:
6 f=500 //frequency in mega hertz
7 f=500*10^6 //frequency in hertz
8 c=3*10^8 //speed of light in m/s
9 Gdmax=1.644 // directivity of a half wave dipole
10 lambda=c/f //wavelength in meter
11 Ae=((lambda)^2*Gdmax)/(4*(%pi)) // Effective area in
    m^2
12 disp(Ae,"effective area of half wave dipole in m^2:"
    )
```

---

**Scilab code Exa 3.6** effective area of hertzian dipole

```
1 // Exa 3.6
2 clc;
3 clear;
4 close;
5 // given :
6 f=100 //frequency in Mhz
7 f=100*10^6 //frequency in hertz
8 c=3*10^8 //speed of light in m/s
9 D=1.5 // directivity
10 lambda=c/f //wavelength in meter
11 Ae=(lambda^2*D)/(4*(%pi)) // effective area in m^2
12 disp(Ae,"Effective area of hertezian dipole in m^2:"
    ")
```

---



# Chapter 4

## ANALYSIS OF LINEAR ARRAYS

Scilab code Exa 4.1 null to null beam width of a broadside array

```
1 //Exa 4.1
2 clc;
3 clear;
4 close;
5 L=poly(0,'L') //Defining L as lambda
6 l=10*L
7 N=20 // number of elements
8 d=1/N
9 // formula : BW=(2*(L/d)*1/N)
10 BW1=(horner((2*L/(N*d)),1))
11 disp(BW1,"Null-to-null BW of broadside array in
        radians when l=10*L,N=20:")
12 l=50*L
13 N=100 // number of elements
14 d=1/N
15 // formula : BW=(2*(L/d)*1/N)
16 BW2=(horner((2*L/(N*d)),1))
17 disp(BW2,"Null-to-null BW of broadside array in
        radians when l=50*L,N=100:")
```

```

18 l=20*L
19 N=50 // number of elements
20 d=1/N
21 // formula : BW=(2*(L/d)*1/N)
22 BW3=(horner((2*L/(N*d)),1))
23 disp(BW3,"Null-to-null BW of broadside array in
        radians when l=20*L,N=50:")

```

---

**Scilab code Exa 4.2** null to null beam width of a endfire array

```

1 //Exa 4.2
2 clc;
3 clear;
4 close;
5 L=poly(0,'L') //Defining L as lambda
6 l=10*L
7 N=20 // number of elements
8 d=1/N
9 // formula : BW=(2*sqrt(2*(L/d)*1/N))
10 BW1=2*sqrt(horner((2*L/(N*d)),1))
11 disp(BW1,"Null-to-null BW of end-fire array in
        radians when l=10*L,N=20:")
12 l=50*L
13 N=100 // number of elements
14 d=1/N
15 // formula : BW=(2*sqrt(2*(L/d)*1/N))
16 BW2=2*sqrt(horner((2*L/(N*d)),1))
17 disp(BW2,"Null-to-null BW of end-fire array in
        radians when l=50*L,N=100:")
18 l=20*L
19 N=50 // number of elements
20 d=1/N
21 // formula : BW=(2*sqrt(2*(L/d)*1/N))
22 BW3=2*sqrt(horner((2*L/(N*d)),1))
23 disp(BW3,"Null-to-null BW of end-fire array in

```

radians when  $l=20*L, N=50$ :")

---

**Scilab code Exa 4.3** null to null beam width and directivity

```
1 //Exa 4.3
2 clc;
3 clear;
4 close;
5 f=6 //frequency in GHz
6 f=6*10^9 //frequency in Hz
7 c=3*10^8 //speed of light in m/s
8 l=10 // array length in meter
9 lambda=c/f //wavelength in meter
10 // formula : BWFN = 2*lambda/l
11 BWFN = 2*(lambda/l) // band width in radians
12 disp(BWFN,"null-to-null Beamwidth of broad side
    array in radians:")
13 D=2*(1/lambda) // directivity
14 disp(D,"Directivity:")
```

---

**Scilab code Exa 4.4** Progressive phase shift and array length

```
1 //Exa 4.4
2 clc;
3 clear;
4 close;
5 //given :
6 f=10 //frequency in Ghz
7 f=10*10^9 //frequency in hertz
8 c=3*10^8 //speed of light in m/s
9 lambda=c/f //wavelength in meter
10 N=50 // number of elements
11 d=0.5*lambda // element spacing in meter
```

```

12 Beta=2*(%pi)/lambda // phase shift
13 alpha=Beta*d // progressive phase shift in radians
14 l=N*d // Array length in meter
15 disp(alpha,"progressive phase shift in radians:")
16 disp(l,"Array length in meter")

```

---

**Scilab code Exa 4.5** null to null and half power beam width and directivity

```

1 //Exa 4.5
2 clc;
3 clear;
4 close;
5 // given :
6 N=100 // no. of elements
7 Lm=poly(0,'Lm') // defining Lm as lambda
8 d=0.5*Lm
9 l=N*d // array length
10 B.W.F.N = 114.6 /(l/Lm) // beam width in degrees
11 B.W.F.N=horner(B.W.F.N,1)
12 disp(B.W.F.N ," null-to-null beamwidth in degrees:")
13 H.P.B.W = B.W.F.N/2 // half power beam width in
    degrees
14 disp(H.P.B.W ," half power beamwidth in degrees:")
15 D1=2*(l/Lm) // directivity of broad side array
16 D1=horner(D1,1)
17 D2=4*(l/Lm) // directivity of end fire array
18 D2=horner(D2,1)
19 disp(D1,"directivity of broad side array:")
20 disp(D2,"directivity of end fire array:")
21
22 // note : answer in the book is mis-printed ,the HPBW
    is not 11.46 it should be 1.146 degrees.
23
24 // note: misprint in second step of part a in book

```

correct is  $l=N*d=100*0.5*\lambda=50*\lambda$

---

**Scilab code Exa 4.9** relative excitation levels

```
1 // Exa 4.9
2 clc;
3 clear;
4 close;
5 // given :
6 // formula : combination(n,r)=(factorial(n))/(
    factorial(r)*factorial(n-r))
7 disp("when n=2")
8 n=2
9 a_0=factorial(1)/factorial(0)*factorial(1) //
    relative excitation level 1
10 a_1=factorial(1)/factorial(1)*factorial(0) //
    relative excitation level 2
11 disp((string(a_0)+" "+string(a_1)),"relative
    excitation levels of binomial array at n=2:")
12 disp("when n=3")
13 n=3
14 a_1=factorial(1)/factorial(1)*factorial(0) //
    relative excitation level 2
15 a_0=2*a_1 // relative excitation level 1
16 disp((string(a_1)+" "+string(a_0)+" "+string(a_1)),"
    relative excitation levels of binomial array at n
    =3:")
```

---

**Scilab code Exa 4.10** basic and actual transmission loss

```
1 // Exa 4.10
2 clc;
3 clear;
```

```

4 close;
5 // given :
6 d=30 //separation distance in meter
7 f=10 //frequency in mega hertz
8 f=10*10^6 //frequency in hertz
9 c=3*10^8 //speed of light in m/s
10 lambda=c/f //wavelength in meter
11 Gt=1.65 //transmitting gain in dB
12 Gr=1.65 //receiving gain in dB
13 // basic transmission loss :
14 // formula : Lb=10*log(((4*(%pi)*d)^2/(lambda)^2))
15 Lb=10*log10((4*(%pi)*d)^2/(lambda)^2) // basic
    transmmision loss in dB
16 disp(Lb,"basic transmmision loss in dB:")
17 // actual transmission loss :
18 La=Lb-Gt-Gr // actual transmisson loss in dB
19 disp(La,"actual transmisson loss in dB:")

```

---

#### Scilab code Exa 4.11 basic transmission loss

```

1 //Exa 4.11
2 clc;
3 clear;
4 close;
5 //when frequency=0.3GHz
6 // given :
7 f=0.3 //frequency in Ghz
8 f=0.3*10^9 //frequency in hertz
9 c=3*10^8 //speed of light in m/s
10 lambda=c/f //wavelength in meter
11 d1=1.6 // in Km
12 d1=1.6*10^3 // in meter
13 // formula : Lb=20*log10((4*(%pi)*d)/(lambda))
14 Lb1=20*log10(4*%pi*d1/lambda) // basic transmission
    loss in dB

```

```

15 disp(Lb1," basic transmission loss in dB when d=1.6Km
    , f=0.3GHz:")
16 d2=16 // in Km
17 d2=16*10^3 // in meter
18 // formula : Lb=20*log10((4*(%pi)*d)/(lambda))
19 Lb2=20*log10(4*%pi*d2/lambda) // basic transmission
    loss in dB
20 disp(Lb2," basic transmission loss in dB when d=16Km,
    f=0.3GHz:")
21 d3=160 // in Km
22 d3=160*10^3 // in meter
23 // formula : Lb=20*log10((4*(%pi)*d)/(lambda))
24 Lb3=20*log10(4*%pi*d3/lambda) // basic transmission
    loss in dB
25 disp(Lb3," basic transmission loss in dB when d=160Km
    , f=0.3GHz:")
26 d4=320 // in Km
27 d4=320*10^3 // in meter
28 // formula : Lb=20*log10((4*(%pi)*d)/(lambda))
29 Lb4=20*log10(4*%pi*d4/lambda) // basic transmission
    loss in dB
30 disp(Lb4," basic transmission loss in dB when d=320Km
    , f=0.3GHz:")
31 // when frequency is 3Ghz
32 // given :
33 f=3 //frequency in Ghz
34 f=3*10^9 //frequency in hertz
35 c=3*10^8 //speed of light in m/s
36 lambda=c/f //wavelength in meter
37 d1=1.6 // in Km
38 d1=1.6*10^3 // in meter
39 // formula : Lb=20*log10((4*(%pi)*d)/(lambda))
40 Lb1=20*log10(4*%pi*d1/lambda) // basic transmission
    loss in dB
41 disp(Lb1," basic transmission loss in dB when d=1.6Km
    , f=3GHz:")
42 d2=16 // in Km
43 d2=16*10^3 // in meter

```

```

44 // formula : Lb=20*log10((4*(%pi)*d)/(lambda))
45 Lb2=20*log10(4*%pi*d2/lambda) // basic transmission
    loss in dB
46 disp(Lb2,"basic transmission loss in dB when d=16Km,
    f=3GHz:")
47 d3=160 // in Km
48 d3=160*10^3 // in meter
49 // formula : Lb=20*log10((4*(%pi)*d)/(lambda))
50 Lb3=20*log10(4*%pi*d3/lambda) // basic transmission
    loss in dB
51 disp(Lb3,"basic transmission loss in dB when d=160Km
    ,f=3GHz:")
52 d4=320 // in Km
53 d4=320*10^3 // in meter
54 // formula : Lb=20*log10((4*(%pi)*d)/(lambda))
55 Lb4=20*log10(4*%pi*d4/lambda) // basic transmission
    loss in dB
56 disp(Lb4,"basic transmission loss in dB when d=320Km
    ,f=3GHz:")

```

---

#### Scilab code Exa 4.12 Actual transmission loss

```

1 //Exa 4.12
2 clc;
3 clear;
4 close;
5 // given :
6 Gt=10 // transmission gain in dB
7 Gr=10 // receiving gain in dB
8 //when frequency=0.3GHz
9 // given :
10 f=0.3 //frequency in Ghz
11 f=0.3*10^9 //frequency in hertz
12 c=3*10^8 //speed of light in m/s
13 lambda=c/f //wavelength in meter

```



```

14 d1=1.6 // in Km
15 d1=1.6*10^3 // in meter
16 // formula : Lb=20*log10((4*(%pi)*d)/(lambda))
17 Lb1=20*log10(4*%pi*d1/lambda) // basic transmission
    loss in dB
18 La1=Lb1-Gt-Gr // Actual transmission loss in dB
19 disp(La1," Actual transmission loss in dB when d=1.6
    Km, f=0.3GHz:")
20 d2=16 // in Km
21 d2=16*10^3 // in meter
22 // formula : Lb=20*log10((4*(%pi)*d)/(lambda))
23 Lb2=20*log10(4*%pi*d2/lambda) // basic transmission
    loss in dB
24 La2=Lb2-Gt-Gr // Actual transmission loss in dB
25 disp(La2," Actual transmission loss in dB when d=16Km
    , f=0.3GHz:")
26 d3=160 // in Km
27 d3=160*10^3 // in meter
28 // formula : Lb=20*log10((4*(%pi)*d)/(lambda))
29 Lb3=20*log10(4*%pi*d3/lambda) // basic transmission
    loss in dB
30 La3=Lb3-Gt-Gr // Actual transmission loss in dB
31 disp(La3," Actual transmission loss in dB when d=160
    Km, f=0.3GHz:")
32 d4=320 // in Km
33 d4=320*10^3 // in meter
34 // formula : Lb=20*log10((4*(%pi)*d)/(lambda))
35 Lb4=20*log10(4*%pi*d4/lambda) // basic transmission
    loss in dB
36 La4=Lb4-Gt-Gr // Actual transmission loss in dB
37 disp(La4," Actual transmission loss in dB when d=320
    Km, f=0.3GHz:")
38 // when frequency is 3Ghz
39 // given :
40 f=3 //frequency in Ghz
41 f=3*10^9 //frequency in hertz
42 c=3*10^8 //speed of light in m/s
43 lambda=c/f //wavelength in meter

```

```

44 d1=1.6 // in Km
45 d1=1.6*10^3 // in meter
46 // formula : Lb=20*log10((4*(%pi)*d)/(lambda))
47 Lb1=20*log10(4*%pi*d1/lambda) // basic transmission
    loss in dB
48 La1=Lb1-Gt-Gr // Actual transmission loss in dB
49 disp(La1," Actual transmission loss in dB when d=1.6
    Km, f=3GHz:")
50 d2=16 // in Km
51 d2=16*10^3 // in meter
52 // formula : Lb=20*log10((4*(%pi)*d)/(lambda))
53 Lb2=20*log10(4*%pi*d2/lambda) // basic transmission
    loss in dB
54 La2=Lb2-Gt-Gr // Actual transmission loss in dB
55 disp(La2," Actual transmission loss in dB when d=16Km
    , f=3GHz:")
56 d3=160 // in Km
57 d3=160*10^3 // in meter
58 // formula : Lb=20*log10((4*(%pi)*d)/(lambda))
59 Lb3=20*log10(4*%pi*d3/lambda) // basic transmission
    loss in dB
60 La3=Lb3-Gt-Gr // Actual transmission loss in dB
61 disp(La3," Actual transmission loss in dB when d=160
    Km, f=3GHz:")
62 d4=320 // in Km
63 d4=320*10^3 // in meter
64 // formula : Lb=20*log10((4*(%pi)*d)/(lambda))
65 Lb4=20*log10(4*%pi*d4/lambda) // basic transmission
    loss in dB
66 La4=Lb4-Gt-Gr // Actual transmission loss in dB
67 disp(La4," Actual transmission loss in dB when d=320
    Km, f=3GHz:")

```

---

Scilab code Exa 4.13 receiving power

```
1 //Exa 4.13
2 clc;
3 clear;
4 close;
5 // given :
6 Wt=15 // radiated power in watt
7 f=60 // in MHz
8 f=60*10^6 // in Hz
9 d=10 // in m
10 c=3*10^8 // in m/s
11 lambda=c/f // in meter
12 Gt=1.64 // transmitting gain in dB
13 Gr=1.64 // receiving gain in dB
14 Wr=(Wt*Gt*Gr*(lambda)^2/(4*(%pi)*d)^2) // receiving
    power in watt
15 disp(Wr*1000,"receiving power in mW:")
```

---

# Chapter 6

## HF VHF AND UHF ANTENNAS

Scilab code Exa 6.1 Designing of a rhombic antenna

```
1 //Exa 6.1
2 clc;
3 clear;
4 close;
5 //given :
6 f=30 // frequency in MHz
7 f=30*10^6 // frequency in Hz
8 c=3*10^8 // speed of light in m/s
9 lambda=c/f // wavelength in meter
10 Delta=30 // angle of elevation in Degrees
11 H=lambda/(4*sind(Delta)) // Rhombic height in m
12 phi=90-Delta // tilt angle in Degrees
13 l=lambda/(2*(cosd(phi)^2)) // wire length in m
14 disp(H," Rhombic height in m:")
15 disp(phi," Tilt angle in Degrees:")
16 disp(l," length of wire in meter:")
```

---

Scilab code Exa 6.2 Designing of a rhombic antenna

```
1 //Exa 6.2
2 clc;
3 clear;
4 close;
5 //given :
6 f=20 //frequency in MHz
7 f=20*10^6 // frequency in Hz
8 c=3*10^8 //speed of light in m/s
9 lambda=c/f //wavelength in meter
10 Delta=10 // angle of elevation in Degrees
11 H=lambda/(4*sind(Delta)) // Rhombic height in m
12 phi=90-Delta // tilt angle in Degrees
13 l=lambda/(2*(cosd(phi)^2)) // wire length in m
14 disp(H," Rhombic height in m:")
15 disp(phi," Tilt angle in Degrees:")
16 disp(l," length of wire in meter:")
```

---

Scilab code Exa 6.3 Designing of a rhombic antenna

```
1 //Exa 6.3
2 clc;
3 clear;
4 close;
5 //given :
6 f=30 //frequency in MHz
7 f=30*10^6 //frequency in Hz
8 c=3*10^8 //speed of light in m/s
9 lambda=c/f //wavelength in meter
10
11 disp(" for Delta=10 degrees")
12
13 Delta1=10 // angle of elevation in Degrees
14 H1=lambda/(4*sind(Delta1)) // Rhombic height in m
```

```

15 phi1=90-Delta1 // tilt angle in Degrees
16 l1=lambda/(2*(cosd(phi1)^2)) // wire length in m
17 disp(H1," Rhombic height in m:")
18 disp(phi1," Tilt angle in Degrees:")
19 disp(l1,"length of wire in meter:")
20
21 disp(" for Delta=15 degrees")
22
23 Delta2=15 // angle of elevation in Degrees
24 H2=lambda/(4*sind(Delta2)) // Rhombic height in m
25 phi2=90-Delta2 // tilt angle in Degrees
26 l2=lambda/(2*(cosd(phi2)^2)) // wire length in m
27 disp(H2," Rhombic height in m:")
28 disp(phi2," Tilt angle in Degrees:")
29 disp(l2,"length of wire in meter:")
30
31 disp(" for Delta=20 degrees")
32
33 Delta3=20 // angle of elevation in Degrees
34 H3=lambda/(4*sind(Delta3)) // Rhombic height in m
35 phi3=90-Delta3 // tilt angle in Degrees
36 l3=lambda/(2*(cosd(phi3)^2)) // wire length in m
37 disp(H3," Rhombic height in m:")
38 disp(phi3," Tilt angle in Degrees:")
39 disp(l3,"length of wire in meter:")
40
41 disp(" for Delta=25 degrees")
42
43 Delta4=25 // angle of elevation in Degrees
44 H4=lambda/(4*sind(Delta4)) // Rhombic height in m
45 phi4=90-Delta4 // tilt angle in Degrees
46 l4=lambda/(2*(cosd(phi4)^2)) // wire length in m
47 disp(H4," Rhombic height in m:")
48 disp(phi4," Tilt angle in Degrees:")
49 disp(l4,"length of wire in meter:")
50
51 disp(" for Delta=30 degrees")
52

```

```

53 Delta5=30 // angle of elevation in Degrees
54 H5=lambda/(4*sind(Delta5)) // Rhombic height in m
55 phi5=90-Delta5 // tilt angle in Degrees
56 l5=lambda/(2*(cosd(phi5)^2)) // wire length in m
57 disp(H5," Rhombic height in m:")
58 disp(phi5," Tilt angle in Degrees:")
59 disp(l5,"length of wire in meter:")
60
61 disp(" for Delta=35 degrees")
62
63 Delta6=35 // angle of elevation in Degrees
64 H6=lambda/(4*sind(Delta6)) // Rhombic height in m
65 phi6=90-Delta6 // tilt angle in Degrees
66 l6=lambda/(2*(cosd(phi6)^2)) // wire length in m
67 disp(H6," Rhombic height in m:")
68 disp(phi6," Tilt angle in Degrees:")
69 disp(l6,"length of wire in meter:")
70
71 disp(" for Delta=40 degrees")
72
73 Delta7=40 // angle of elevation in Degrees
74 H7=lambda/(4*sind(Delta7)) // Rhombic height in m
75 phi7=90-Delta7 // tilt angle in Degrees
76 l7=lambda/(2*(cosd(phi7)^2)) // wire length in m
77 disp(H7," Rhombic height in m:")
78 disp(phi7," Tilt angle in Degrees:")
79 disp(l7,"length of wire in meter:")

```

---

#### Scilab code Exa 6.4 Design parameters of rhombic antena

```

1 //Exa 6.4
2 clc;
3 clear;
4 close;
5 // given :

```

```

6 f=30 //frequency in MHz
7 f=30*10^6 //frequency in Hz
8 c=3*10^8 //speed of light in m/s
9 K=0.74 // constant
10 lambda=c/f // in meter
11 Delta=30 // angle of elevation in Degrees
12 H=lambda/(4*sind(Delta)) // Rhombic height in m
13 phi=90-Delta // tilt angle in Degrees
14 l=(lambda/(2*(cosd(phi)^2)))*K // wire length in m
15 disp(H," Rhombic height in m:")
16 disp(phi," Tilt angle in Degrees:")
17 disp(l," length of wire in meter:")

```

---

#### Scilab code Exa 6.5 Design parameters of rhombic antena

```

1 //Exa 6.5
2 clc;
3 clear;
4 close;
5 // given :
6 f=20 //ferquency in MHz
7 f=20*10^6 //frequency in Hz
8 c=3*10^8 //speed of light in m/s
9 K=0.74 // constant
10 lambda=c/f //wavelength in meter
11 Delta=20 // angle of elevation in Degrees
12 H=lambda/(4*sind(Delta)) // Rhombic height in m
13 phi=90-Delta // tilt angle in Degrees
14 l=(lambda/(2*(cosd(phi)^2)))*K // wire length in m
15 disp(H," Rhombic height in m:")
16 disp(phi," Tilt angle in Degrees:")
17 disp(l," length of wire in meter:")

```

---



**Scilab code Exa 6.6** Design a three element yagi uda antenna

```
1 //Exa 6.6
2 clc;
3 clear;
4 close;
5 // given :
6 f_MHz=172 // frequency in MHz
7 c=3*10^8 // speed of light in m/s
8 lambda=c/f_MHz // wavelength in m
9 La=478/f_MHz // length of driven element in feet
10 Lr=492/f_MHz // length of reflector in feet
11 Ld=461.5/f_MHz // length of director in feet
12 S=142/f_MHz // element spacing in feet
13 disp(La,"length of driven element in feet:")
14 disp(Lr,"length of reflector in feet:")
15 disp(Ld,"length of director in feet:")
16 disp(S,"element spacing in feet:")
```

---

**Scilab code Exa 6.7** Designing of a six element yagi uda antenna

```
1 //Exa6.7
2 clc;
3 clear;
4 close;
5 // given :
6 G=12 // required gain in dB
7 f=200 // frequency in MHz
8 f=200*10^6 // frequency in Hz
9 c=3*10^8 // speed of light in m/s
10 lambda=c/f // wavelength in m
11 La=0.46*lambda // length of driven element in m (
    note: in book La is given 0.416*lambda misprint)
12 Lr=0.475*lambda // length of reflector in m
13 Ld1=0.44*lambda // length of director1 in m
```

```

14 Ld2=0.44*lambda // length of director2 in m
15 Ld3=0.43*lambda // length of director3 in m
16 Ld4=0.40*lambda // length of director4 in m
17 SL=0.25*lambda // spacing between reflector and
    driver in m
18 Sd=0.31*lambda // spacing director and driving
    element in m
19 d=0.01*lambda // diameter of elements in m
20 l=1.5*lambda // length of array in m
21 disp(La,"length of driven element in m:")
22 disp(Lr,"length of reflector in m:")
23 disp(Ld1,"length of director1 in m:")
24 disp(Ld2,"length of director2 in m:")
25 disp(Ld3,"length of director3 in m:")
26 disp(Ld4,"length of director4 in m:")
27 disp(SL,"spacing between reflector and driver in m:"
    )
28 disp(Sd,"spacing director and driving element in m:"
    )
29 disp(d,"diameter of elements in m:")
30 disp(l,"length of array in m:")

```

---

### Scilab code Exa 6.8 Designing of a long periodic antenna

```

1 //Exa 6.8
2 clc;
3 clear;
4 close;
5 // given :
6 G=9 // required gain in dB
7 f_l=125 // lowest frequency in MHz
8 f_l=125*10^6 // lowest frequency in Hz
9 f_h=500 // highest frequency in MHz
10 f_h=500*10^6 // lowest frequency in Hz
11 c=3*10^8 // speed of light in m/s

```

```

12 lambda_l=c/f_l // longest wavelength in m
13 lambda_s=c/f_h // shortest wavelength in m
14 tau=0.861 // scaling factor
15 sigma=0.162 // spacing factor
16 alpha=2*atand((1-tau)/(4*sigma)) // wedge angle in
    Degrees
17 L1=lambda_l/2 // in m
18 L2=tau*L1 // in m
19 L3=tau*L2 // in m
20 L4=tau*L3 // in m
21 L5=tau*L4 // in m
22 L6=tau*L5 // in m
23 L7=tau*L6 // in m
24 L8=tau*L7 // in m
25 L9=tau*L8 // in m
26 L10=tau*L9 // in m
27 L11=tau*L10 // in m
28
29 // element spacing relation
30 //formula : sn=2*sigma*Ln
31 S1=2*sigma*L1 // in m
32 S2=2*sigma*L2 // in m
33 S3=2*sigma*L3 // in m
34 S4=2*sigma*L4 // in m
35 S5=2*sigma*L5 // in m
36 S6=2*sigma*L6 // in m
37 S7=2*sigma*L7 // in m
38 S8=2*sigma*L8 // in m
39 S9=2*sigma*L9 // in m
40 S10=2*sigma*L10 // in m
41 S11=2*sigma*L11 // in m
42
43
44 disp("designing of log-periodic antenna:")
45
46 disp(L1,"L1 in m:")
47 disp(L2,"L2 in m:")
48 disp(L3,"L3 in m:")

```

```

49 disp(L4,"L4 in m:")
50 disp(L5,"L5 in m:")
51 disp(L6,"L6 in m:")
52 disp(L7,"L7 in m:")
53 disp(L8,"L8 in m:")
54 disp(L9,"L9 in m:")
55 disp(L10,"L10 in m:")
56 disp(L11,"L11 in m:")
57
58 disp("elements spacing relation:")
59 disp(S1,"S1 in m:")
60 disp(S2,"S2 in m:")
61 disp(S3,"S3 in m:")
62 disp(S4,"S4 in m:")
63 disp(S5,"S5 in m:")
64 disp(S6,"S6 in m:")
65 disp(S7,"S7 in m:")
66 disp(S8,"S8 in m:")
67 disp(S9,"S9 in m:")
68 disp(S10,"S10 in m:")
69 disp(S11,"S11 in m:")

```

---

**Scilab code Exa 6.9** induced voltage in a loop antenna

```

1 //Exa 6.9
2 clc;
3 clear;
4 close;
5 // given :
6 E_rms=10 // electric field in mV/m
7 E_rms=10*10^-3 // electric field in V/m
8 f=2 // frequency in MHz
9 f=2*10^6 // frequency in Hz
10 N=10 // number of turns
11 phi=0 // angle between the plane of loop and

```

```

        direction of incident wave in Degrees
12 S=1.4 // area of loop antenna in m^2
13 c=3*10^8 // speed of light in m/s
14 lambda=c/f // wavelength in m
15 E_max=sqrt(2)*E_rms // electric field in V/m
16 V_rms=(2*pi*E_max*S*N/lambda)*cosd(phi) // induced
    voltage
17 disp(V_rms*1000,"induced voltage in mV:")

```

---

**Scilab code Exa 6.10** radiation resistance of a loop antenna

```

1 //Exa6.10
2 clc;
3 clear;
4 close;
5 //given :
6 D=0.5 // diameter of loop antenna in m
7 a=D/2 // radius of loop antenna in m
8 f=1 // frequency in MHz
9 f=1*10^6 // frequency in Hz
10 c=3*10^8 // speed of light in m/s
11 lambda=c/f // wavelength in m
12 Rr=3720*(a/lambda) // radiation resistance of loop
    antenna in ohm
13 disp(Rr,"radiation resistance of loop antenna in ohm
    :")

```

---

**Scilab code Exa 6.11** Directivity of a loop antenna

```

1 //Exa 6.11
2 clc;
3 clear;
4 close;

```

```

5 //given :
6 a=0.5 // radius of loop antenna in m
7 f=0.9 // frequency in MHz
8 f=0.9*10^6 // frequency in Hz
9 c=3*10^8 // speed of light in m/s
10 lambda=c/f // wavelength in m
11 k=(2*%pi*a)/lambda // constant
12 disp("the value of k is:")
13 disp(k)
14 disp("since ,k<1/3")
15 disp("So Directivity of loop antenna is D=1.5")

```

---

**Scilab code Exa 6.13** Directivity and radiation resistance of a loop antenna

```

1 //Exa 6.13
2 clc;
3 clear;
4 close;
5 Lm=poly(0, 'Lm') // defining Lm as lambda
6 d=1.5*Lm // diameter of antenna in m
7 a=d/2 // radius of antenna in m
8 // formula : Rr=3720*(a/Lm)
9 Rr=3720*(a/Lm) // radiation resistance of loop
   antenna in ohm
10 Rr=horner(Rr,1)
11 // formula : D=4.25*(a/Lm)
12 D=4.25*(a/Lm) // Directivity of the loop antenna
13 D=horner(D,1)
14 disp(Rr,"radiation resistance of the loop antenna in
   ohm:")
15 disp(D,"Directivity of the loop antenna:")

```

---

**Scilab code Exa 6.14** array length number of elements and null to null beam width

```
1 //Exa 6.14
2 clc;
3 clear;
4 close;
5 //given :
6 Gp=28 // power gain
7 Lm=poly(0, 'Lm') // defining Lm as lambda
8 d=Lm/2 // length of dipole
9 //formula : Gp=4*(L/lambda)
10 L=Gp*Lm/4 // array length
11 disp(L,"array length(where Lm is wavelength in m):")
12 N=7*2 // Number of elements in the array when spaced
    at lambda/2
13 disp(N,"Number of elements in the array when spaced
    at lambda/2:")
14 // formula : B.W=2*sqrt((2*/N)*(lambda/d))
15 BW=2*sqrt(horner((2*Lm/(N*d)),1)) // null-to-null
    beam width in radians
16 BW_d=BW*180/%pi // null-to-null beam width in
    degrees
17 disp(BW_d,"null-to-null beam width in degrees:")
18
19
20
21 // Answer of null-to-null beam width in degrees is
    rounded-off in book.
```

---

**Scilab code Exa 6.15** null to null and half power beam width and directivity

```
1 //Exa 6.15
2 clc;
```

```

3 clear;
4 close;
5 // given :
6 S=0.05 // spacing in m
7 Dh=0.1 // diameter of helical antenna in m
8 N=20 // number of turns
9 f=1000 // frequency in MHz
10 f=1000*10^6 // frequency in MHz
11 c=3*10^8 // speed of light in m/s
12 lambda=c/f // wavelength in m
13 C=%pi*Dh // circumference of helix in m
14 La=N*S // axial length in m
15 phi_not=(115*(lambda^(3/2))/(C*sqrt(La))) // B.W.F.N
    ., null-to-null beamwidth of main beam in
    Degrees
16 phi=(52*lambda^(3/2)/(C*sqrt(La))) // H.P.B.W, half
    power beamwidth in Degrees
17 D=(15*N*C^2*S/(lambda)^3) // Directivity
18 disp(phi_not,"B.W.F.N., null-to-null beamwidth of
    main beam in Degrees:")
19 disp(phi,"H.P.B.W, half power beamwidth in Degrees:"
    )
20 disp(D," Directivity:")

```

---



# Chapter 7

## MICROWAVE ANTENNAS

Scilab code Exa 7.1 null to null and half power beam width of a paraboloid reflector

```
1 //Exa 7.1
2 clc;
3 clear;
4 close;
5 // given :
6 D=2 // Diameter of paraboloid reflector in m
7 c=3*10^8 // speed of light in m/s
8 f=5 // frequency in GHz
9 f=5*10^9 // frequency in Hz
10 lambda=c/f // wavelength in m
11 BWFN=140*(lambda/D) // null-to-null beamwidth in
    degrees
12 HPBW=70*(lambda/D) // half power beamwidth in
    degrees
13 disp(BWFN,"null-to-null beamwidth in degrees:")
14 disp(HPBW,"half power beamwidth in degrees:")
```

---

Scilab code Exa 7.2 gain of the paraboloid reflector antenna

```

1 //Exa 7.2
2 clc;
3 clear;
4 close;
5 // given :
6 D=2 // mouth diameter of paraboloid reflector in m
7 c=3*10^8 // speed of light in m/s
8 f=5 // frequency in GHz
9 f=5*10^9 // frequency in Hz
10 lambda=c/f // wavelength in m
11 G=6.4*(D/lambda)^2 // power gain of paraboloid
12 G_p=10*log10(G) //power gain in dB
13 disp(G_p,"power gain in dB:")

```

---

**Scilab code Exa 7.3** band width between first null and half power points

```

1 //Exa 7.3
2 clc;
3 clear;
4 close;
5 // given :
6 D_a=0.15 // mouth Diameter of paraboloid in m
7 c=3*10^8 // speed of light in m/s
8 f=10 // frequency in GHz
9 f=10*10^9 // frequency in Hz
10 lambda=c/f // wavelength in m
11 BWFN=140*(lambda/D_a) // null-to-null beamwidth in
    degrees
12 HPBW=70*(lambda/D_a) // half power beamwidth in
    degrees
13 disp(BWFN,"null-to-null beamwidth in degrees:")
14 disp(HPBW,"half power beamwidth in degrees:")
15 G_p=6.4*(D_a/lambda)^2 // power gain of paraboloid
16 G_p=10*log10(G_p) // power gain in dB
17 disp(G_p,"power gain in dB")

```

---

**Scilab code Exa 7.4** Power gain in Decibels

```
1 //Exa 7.4
2 clc;
3 clear;
4 close;
5 // given :
6 D_a=1.8 //mouth diameter of paraboloid reflector in
      m
7 c=3*10^8 // speed of light in m/s
8 f=2 // frequency in GHz
9 f=2*10^9 // frequency in Hz
10 lambda=c/f // wavelength in m
11 G_p=6.4*(D_a/lambda)^2 // power gain of paraboloid
12 G_p=10*log10(G_p) // power gain in dB
13 disp(G_p,"power gain in dB")
```

---

**Scilab code Exa 7.5** mouth diameter HPBW and power gain

```
1 //Exa 7.5
2 clc;
3 clear;
4 close;
5 // given :
6 c=3*10^8 // speed of light in m/s
7 f=5 // frequency in GHz
8 f=5*10^9 // frequency in Hz
9 lambda=c/f // wavelength in m
10 BWFN=10 // null-to-null beamwidth in degrees
11 // formula: BWFN=140*(lambda/D_a)
12 D_a=140*lambda/BWFN // mouth Diameter of paraboloid
      reflector in m
```

```

13 disp(D_a,"mouth Diameter of paraboloid reflector in
    m:")
14 HPBW=70*(lambda/D_a) // half power beamwidth in
    degrees
15 disp(HPBW,"half power beamwidth in degrees:")
16 G_p=6.4*(D_a/lambda)^2 // power gain of paraboloid
17 disp(G_p,"power gain of paraboloid:")

```

---

**Scilab code Exa 7.6** beam width directivity and capoture area

```

1 //Exa 7.6
2 clc;
3 clear;
4 close;
5 // given :
6 b=0.65 // illumination efficiency
7 D_a=6 // mouth diameter of paraboloid reflector in m
8 c=3*10^8 // speed of light in m/s
9 f=10 // frequency in GHz
10 f=10*10^9 // frequency in Hz
11 lambda=c/f // wavelength in m
12 A=%pi*(D_a)^2/4 // Actual area in m^2
13 A_c=0.65*A // capture area in m^2
14 D=6.4*(D_a/lambda)^2 // directivity
15 D=10*log10(D) // directivity in dB
16 phi=70*(lambda/D_a) // half power beam width in
    degrees
17 phi_not=2*phi // null-to-null main beam width in
    degrees
18 disp(D,"directivity in dB:")
19 disp(phi,"half power beam width in degrees:")
20 disp(phi_not,"null-to-null main beam width in
    degrees:")
21 disp(A_c,"capture area in m^2:")

```

---

Scilab code Exa 7.7 minimum distance required between two antennas

```
1 //Exa 7.7
2 clc;
3 clear;
4 close;
5 // given :
6 D_a=6 // Diameter of paraboloid reflector in m
7 c=3*10^8 // speed of light in m/s
8 f=4 // frequency in GHz
9 f=4*10^9 // frequency in Hz
10 lambda=c/f // wavelength in m
11 r=2*D_a^2/lambda // required minimum distance
    between two antennae in m
12 disp(r,"required minimum distance between two
    antennae in m:")
```

---

Scilab code Exa 7.8 mouth diameter and beam width

```
1 //Exa 7.8
2 clc;
3 clear;
4 close;
5 // given :
6 G_p=1000 // gain
7 c=3*10^8 // speed of light in m/s
8 f=3 // frequency in GHz
9 f=3*10^9 // frequency in Hz
10 lambda=c/f // wavelength in m
11 // formula :  $G_p=6.4*(D_a/\lambda)^2$  // power gain
12 D_a=lambda*(sqrt(G_p/6.4)) // mouth Diameter of
    paraboloid in m
```

```

13 BWFN=140*(lambda/D_a) // null-to-null beamwidth in
    degrees
14 HPBW=70*(lambda/D_a) // half power beamwidth in
    degrees
15 disp(D_a,"mouth Diameter of paraboloid in m")
16 disp(BWFN,"null-to-null beamwidth in degrees:")
17 disp(HPBW,"half power beamwidth in degrees:")

```

---

**Scilab code Exa 7.9** capture area and beam width of paraboloid antenna

```

1 //Exa 7.9
2 clc;
3 clear;
4 close;
5 // given :
6 c=3*10^8 // speed of light in m/s
7 f=10 // frequency in GHz
8 f=10*10^9 // frequency in Hz
9 lambda=c/f // wavelength in m
10 G_p=75 // power gain in dB
11 // formula : G_p=10*log10(G_p) // power gain in dB
12 G=10^(G_p/10) // simple power gain
13 // formula : G=6.4*(D_a/lambda)^2 // power gain
14 D_a=lambda*(sqrt(G/6.4)) // mouth Diameter of
    paraboloid in m
15 A=%pi*(D_a)^2/4 // Actual area in m^2
16 A_c=0.65*A // capture area in m^2
17 BWFN=140*(lambda/D_a) // null-to-null beamwidth in
    degrees
18 HPBW=70*(lambda/D_a) // half power beamwidth in
    degrees
19 disp(BWFN,"null-to-null beamwidth in degrees:")
20 disp(HPBW,"half power beamwidth in degrees:")
21 disp(A_c,"capture area in m^2:")
22

```

```
23
24
25 //note : answer of A_c in book is 2269.83 m^2 but
    in scilab 2270.20 m^2
```

---

#### Scilab code Exa 7.10 HPBW BWFN and power gain

```
1 //Exa 7.10
2 clc;
3 clear;
4 close;
5 // given :
6 D_a=60 // mouth diameter of paraboloid reflector in
    m
7 c=3*10^8 // speed of light in m/s
8 f=2 // frequency in GHz
9 f=2*10^9 // frequency in Hz
10 lambda=c/f // wavelength in m
11 phi=70*(lambda/D_a) // half power beam width in
    degrees
12 phi_not=140*(lambda/D_a) // null-to-null main beam
    width in degrees
13 disp(phi,"half power beam width in degrees:")
14 disp(phi_not,"null-to-null main beam width in
    degrees:")
15 G_p=6.4*(D_a/lambda)^2 // power gain of paraboloid
16 G_p=10*log10(G_p) //power gain in dB
17 disp(G_p,"power gain in dB:")
```

---

#### Scilab code Exa 7.11 power gain

```
1 //Exa 7.11
2 clc;
```

```

3 clear;
4 close;
5 // given :
6 D=22 // mouth diameter of paraboloid reflector in m
7 c=3*10^8 // speed of light in m/s
8 f=5 // frequency in GHz
9 f=5*10^9 // frequency in Hz
10 lambda=c/f // wavelength in m
11 b=0.6 // illumination efficiency
12 G_p=b*(D/lambda)^2 // power gain of paraboloid
13 G_p=10*log10(G_p) //power gain in dB
14 disp(G_p,"power gain in dB:")

```

---

**Scilab code Exa 7.12** mouth diameter and capture area of a paraboloid antenna

```

1 //Exa 7.12
2 clc;
3 clear;
4 close;
5 // given :
6 c=3*10^8 // speed of light in m/s
7 f=2 // frequency in GHz
8 f=2*10^9 // frequency in Hz
9 lambda=c/f // wavelength in m
10 BWFN=12 // null-to-null main beam width in degrees
11 // formula : BWFN=140*(lambda/D_a)
12 D_a=140*lambda/BWFN // mouth diameter of paraboloid
    reflector in m
13 A=%pi*(D_a)^2/4 // Actual area in m^2
14 A_c=0.65*A // capture area in m^2
15 disp(D_a,"mouth diameter of paraboloid reflector in
    m:")
16 disp(A_c,"capture area in m^2:")

```

---



**Scilab code Exa 7.13** mouth diameter and power gain of paraboloid reflector antenna

```
1 //Exa 7.13
2 clc;
3 clear;
4 close;
5 // given :
6 c=3*10^8 // speed of light in m/s
7 f=2.5 // frequency in GHz
8 f=2.5*10^9 // frequency in Hz
9 lambda=c/f // wavelength in m
10 BWFN=3 // null-to-null main beam width in degrees
11 // formula : BWFN=140*(lambda/D_a)
12 D_a=140*lambda/BWFN // mouth diameter of paraboloid
    reflector in m
13 G=6.4*(D_a/lambda)^2 // power gain of paraboloid
14 G_p=10*log10(G) //power gain in dB
15 disp(G_p,"power gain in dB:")
16 disp(D_a,"mouth diameter of paraboloid reflector in
    m:")
```

---

**Scilab code Exa 7.14** null to null beam width and power gain of paraboloid reflector antenna

```
1 // Exa7.14
2 clc;
3 clear;
4 close;
5 // given :
6 phi=5 // HPBW, half power beam width in Degrees
```

```

7 phi_not=2*phi // BWFN, null-to-null beam width in
  degrees
8 Lm=poly(0, 'Lm') // defining Lm as lambda
9 // formula : phi=70*(Lm/D_a) // where Lm is
  wavelength in m and D_a is mouth diameter in m
10 D_a=(70*Lm)/phi
11 G_p=6.4*(D_a/Lm)^2
12 G_p=horner(G_p,1)
13 G_p=10*log10(G_p) // power gain in dB
14 disp(phi_not, "BWFN, null-to-null beam width in
  degrees:")
15 disp(G_p, "power gain in dB:")

```

---

**Scilab code Exa 7.15** Power gain of a paraboloid reflector antenna

```

1 // Exa 7.15
2 clc;
3 clear;
4 close;
5 Lm=poly(0, 'Lm') // defining Lm as lambda
6 D_a=8*Lm // where D_a is mouth diameter in m and Lm
  is wavelength in m
7 // formula : G_p=6.4*(D/lambda)^2
8 G_p=6.4*(D_a/Lm)^2 //power gain
9 G_p=horner(G_p,1)
10 G_p=10*log10(G_p) // power gain in dB
11 disp(G_p, "power gain in dB:")

```

---

**Scilab code Exa 7.16** null to null and half power beam width and directivity

```

1 //Exa 7.16
2 clc;

```

```

3 clear;
4 close;
5 Lm=poly(0,'Lm') // defining Lm as lambda
6 D_a=6*Lm // where D_a is mouth diameter in m and Lm
   is wavelength
7 // formula : HPBW=phi=70*(lambda/D_a)
8 phi=70*(Lm/D_a) // half power beam width in degrees
9 phi=horner(phi,1)
10 phi_not=2*phi // null-to-null beam width in degrees
11 // formula : D=6.4*(D_a/lambda)^2
12 D=6.4*(D_a/Lm)^2
13 D=horner(D,1)
14 disp(D," Directivity:")
15 disp(phi," half power beam width in degrees:")
16 disp(phi_not," null-to-null beam width in degrees:")

```

---

**Scilab code Exa 7.17** beam width power gain and directivity

```

1 //Exa 7.17
2 clc;
3 clear;
4 close;
5 // given :
6 f=6 // frequency in GHz
7 f=6*10^9 // frequency in Hz
8 c=3*10^8 // speed of light in m/s
9 lambda=c/f // wavelength in m
10 d=12 // aperture length in cm
11 d=12*10^-2 // aperture length in m
12 w=6 // aperture width in cm
13 w=6*10^-2 // aperture width in m
14 phi_E=56*(lambda/d) // half power beam width for
   aperture length d in Degrees
15 phi_H=67*(lambda/w) // half power beam width for
   aperture width w in Degrees

```

```

16 G_p=(4.5*w*d)/(lambda)^2 // power gain
17 G_p=10*log10(G_p) // power gain in dB
18 D=(7.5*w*d)/(lambda)^2 // Directivity
19 disp(phi_E,"half power beam width for aperture
    length d in Degrees:")
20 disp(phi_H,"half power beam width for aperture width
    w in Degrees:")
21 disp(G_p,"power gain in dB:")
22 disp(D,"Directivity:")

```

---

**Scilab code Exa 7.18** power gain of a square horn antenna

```

1 // Exa 7.18
2 clc;
3 clear;
4 close;
5 Lm=poly(0,'Lm') // defining Lm as lambda
6 d=8*Lm // where d is aperture length and Lm is
    wavelength
7 w=8*Lm // where w is aperture width
8 //formula : G_p=(4.5*w*d)/lambda^2
9 G_p=(4.5*w*d)/Lm^2 // power gain
10 G_p=horner(G_p,1)
11 G_p=10*log10(G_p) // power gain in dB
12 disp(G_p,"power gain in dB:")

```

---

**Scilab code Exa 7.19** power gain and directivity of a horn antenna

```

1 //Exa 7.19
2 clc;
3 clear;
4 close;
5 // given :

```

```

6 f=6 // frequency in GHz
7 f=6*10^9 // frequency in Hz
8 c=3*10^8 // speed of light in m/s
9 lambda=c/f // wavelength in m
10 d=10 // aperture length in cm
11 d=10*10^-2 // aperture length in m
12 w=5 // aperture width in cm
13 w=5*10^-2 // aperture width in m
14 G_p=(4.5*w*d)/(lambda)^2 // power gain
15 G_p=10*log10(G_p) // power gain in dB
16 D=(7.5*w*d)/(lambda)^2 // Directivity
17 D=10*log10(D) // directivity in dB
18 disp(G_p,"power gain in dB:")
19 disp(D,"Directivity in dB:")

```

---

#### Scilab code Exa 7.20 complementary slot impedances

```

1 //Exa 7.20
2 clc;
3 clear;
4 close;
5 // given :
6 eta_0=377 //intrinsic impedance in ohm
7 disp("when Zd=73+%i*42.5")
8 Zd=73+%i*42.5 // dipole impedance
9 // formula : zs*zd=(eta_0)^2/4
10 Zs=eta_0^2/(4*Zd) // slot impedance in ohm
11 disp(Zs,"complementary slot impedance in ohm:")
12
13 disp("when Zd=67+%i*0")
14 Zd=67+%i*0 // dipole impedance
15 // formula : zs*zd=(eta_0)^2/4
16 Zs=eta_0^2/(4*Zd) // slot impedance in ohm
17 disp(Zs,"complementary slot impedance in ohm:")
18

```

```

19 disp("when Zd=710+%i*0")
20 Zd=710+%i*0 // dipole impedance
21 // formula : zs*zd=(eta_0)^2/4
22 Zs=eta_0^2/(4*Zd) // slot impedance in ohm
23 disp(Zs,"complementary slot impedance in ohm:")
24
25
26 disp("when Zd=500+%i*0")
27 Zd=500+%i*0 // dipole impedance
28 // formula : zs*zd=(eta_0)^2/4
29 Zs=eta_0^2/(4*Zd) // slot impedance in ohm
30 disp(Zs,"complementary slot impedance in ohm:")
31
32
33 disp("when Zd=50+%i*20")
34 Zd=50+%i*20 // dipole impedance
35 // formula : zs*zd=(eta_0)^2/4
36 Zs=eta_0^2/(4*Zd) // slot impedance in ohm
37 disp(Zs,"complementary slot impedance in ohm:")
38
39
40 disp("when Zd=50-%i*25")
41 Zd=50-%i*25 // dipole impedance
42 // formula : zs*zd=(eta_0)^2/4
43 Zs=eta_0^2/(4*Zd) // slot impedance in ohm
44 disp(Zs,"complementary slot impedance in ohm:")
45
46
47 disp("when Zd=300+%i*0")
48 Zd=300+%i*0 // dipole impedance
49 // formula : zs*zd=(eta_0)^2/4
50 Zs=eta_0^2/(4*Zd) // slot impedance in ohm
51 disp(Zs,"complementary slot impedance in ohm:")

```

---

# Chapter 9

## WAVE PROPAGATION

Scilab code Exa 9.1 Required transmitter power

```
1 //Exa 9.1
2 clc;
3 clear;
4 close;
5 // given :
6 f=1.7 //frequency in MHz
7 f=1.7*10^6 //frequency in Hz
8 E=0.5 //electric field in mV/m
9 E=0.5*10^(-3) //electric field in V/m
10 sigma=5*10^-5 // conductivity in mho/cm
11 eta=0.5 // efficiency
12 epsilon_r=10 //relative permittivity
13 c=3*10^8 // speed of light in m/s
14 lambda=c/f //wavelength in m
15 d=10 // distance in km
16 d=10*10^3 // distance in m
17 Df=1.8*10^12*(sigma/f) // dissipation factor
18 b=atand((epsilon_r+1)/Df) // phase constant in
    degrees
19 p=(%pi/Df)*(d/lambda)*cosd(b) // numerical distance
20 A=((2+0.3*p)/(2+p+0.6*p^2))-(sind(b)*(sqrt(p/2)*exp
```

```

        ((-5*p)/8))) // factor of the ground losses
21 // formula: E=(A*E_not)/d
22 //E_not=300*sqrt(P_kW) // E_not is in mV/m
23 //E_not=300*sqrt(P_kW)*10^(-3) // E_not is in V/m
24 P_kW=(E*d*10^-3/(A*300*10^-3))^2 // Power in kW and
    d is in km
25 // formula : P=Ptx*eta
26 Ptx=P_kW/eta //transmitter power required in kW
27 disp(Ptx*1000,"transmitter power required in W:")
28
29
30 //note: answer in the book is wrong.scilab give Ptx
    =11.48W whereas answer in the book is Ptx=12.6W

```

---

**Scilab code Exa 9.2** field strength of the ground wave

```

1 //Exa 9.2
2 clc;
3 clear;
4 close;
5 // given :
6 disp("for frequency=500kHz")
7 f1=500*10^3 //frequency in Hz
8 c=3*10^8 //speed of light in m/s
9 lambda1=c/f1 // wavelength in m
10 d=100 //distance in km
11 d=100*10^3 //distance in m
12 Pt=100 // transmitter power in kW
13 eta=50 // efficiency in percentage
14 eta=0.5 // efficiency
15 P_kW=Pt*eta // radiated power in kW
16 E_not=300*1.28*sqrt(P_kW) // field strength in mV/m
    without ground losses at 1km
17
18 // for sea water earth

```



```

19 epsilon_r=81 // permittivity
20 sigma=45*10^-3 // conductivity in mho/cm
21 Df=1.8*10^12*(sigma/f1) // dissipation factor
22 b=atand((epsilon_r+1)/Df) // phase constant in
    degrees
23 p=(%pi/Df)*(d/lambda1)*cosd(b) // numerical distance
24 A=((2+0.3*p)/(2+p+0.6*p^2))-(sind(b)*(sqrt(p/2)*exp
    ((-5*p)/8))) // factor of the ground losses
25 E1=(A*E_not)/(d) // in V/m
26 disp(E1*1000,"electric field strength for sea water
    earth in mV/m when f=500kHz:")
27
28 // for good soil
29 epsilon_r=20 // permittivity
30 sigma=10^-4 // conductivity in mho/cm
31 Df=1.8*10^12*(sigma/f1) // dissipation factor
32 b=atand((epsilon_r+1)/Df) // phase constant in
    degrees
33 p=(%pi/Df)*(d/lambda1)*cosd(b) // numerical distance
34 A=((2+0.3*p)/(2+p+0.6*p^2))-(sind(b)*(sqrt(p/2)*exp
    ((-5*p)/8))) // factor of the ground losses
35 E2=(A*E_not)/(d) // in V/m
36 disp(E2*1000,"electric field strength for good soil
    in mV/m when f=500kHz:")
37
38 // for poor soil
39 epsilon_r=10 // permittivity
40 sigma=0.2*10^-4 // conductivity in mho/cm
41 Df=1.8*10^12*(sigma/f1) // dissipation factor
42 b=atand((epsilon_r+1)/Df) // phase constant in
    degrees
43 p=(%pi/Df)*(d/lambda1)*cosd(b) // numerical distance
44 A=((2+0.3*p)/(2+p+0.6*p^2))-(sind(b)*(sqrt(p/2)*exp
    ((-5*p)/8))) // factor of the ground losses
45 E3=(A*E_not)/(d) // in V/m
46 disp(E3*1000,"electric field strength for poor soil
    in mV/m when f=500kHz:")
47

```

```

48 // for cities , industrial areas
49 epsilon_r=5 // permittivity
50 sigma=10^-5 // conductivity in mho/cm
51 Df=1.8*10^12*(sigma/f1) // dissipation factor
52 b=atand((epsilon_r+1)/Df) // phase constant in
    degrees
53 p=(%pi/Df)*(d/lambda1)*cosd(b) // numerical distance
54 A=((2+0.3*p)/(2+p+0.6*p^2))-(sind(b)*(sqrt(p/2)*exp
    ((-5*p)/8))) // factor of the ground losses
55 E4=(A*E_not)/(d) // in V/m
56 disp(E4*1000,"electric field strength for cities ,
    industrial areas in mV/m when f=500kHz:")
57
58 // for Rocky soil , flat sandy
59 epsilon_r=10 // permittivity
60 sigma=2*10^-3 // conductivity in mho/cm
61 Df=1.8*10^12*(sigma/f1) // dissipation factor
62 b=atand((epsilon_r+1)/Df) // phase constant in
    degrees
63 p=(%pi/Df)*(d/lambda1)*cosd(b) // numerical distance
64 A=((2+0.3*p)/(2+p+0.6*p^2))-(sind(b)*(sqrt(p/2)*exp
    ((-5*p)/8))) // factor of the ground losses
65 E5=(A*E_not)/(d) // in V/m
66 disp(E5*1000,"electric field strength for Rocky
    soil , flat sandy in mV/m when f=500kHz:")
67
68 // for medium hills , forestation
69 epsilon_r=13 // permittivity
70 sigma=5*10^-5 // conductivity in mho/cm
71 Df=1.8*10^12*(sigma/f1) // dissipation factor
72 b=atand((epsilon_r+1)/Df) // phase constant in
    degrees
73 p=(%pi/Df)*(d/lambda1)*cosd(b) // numerical distance
74 A=((2+0.3*p)/(2+p+0.6*p^2))-(sind(b)*(sqrt(p/2)*exp
    ((-5*p)/8))) // factor of the ground losses
75 E6=(A*E_not)/(d) // in V/m
76 disp(E6*1000,"electric field strength for medium
    hills , forestation in mV/m when f=500kHz:")

```

```

77
78
79
80 disp(" for frequency=1500kHz")
81
82 f2=1500 // in kHz
83 f2=1500*10^(3) // in Hz
84 lambda2=c/f2 // in m
85 // for sea water earth
86 epsilon_r=81 // permittivity
87 sigma=45*10^-3 // conductivity in mho/cm
88 Df=1.8*10^12*(sigma/f2) // dissipation factor
89 b=atand((epsilon_r+1)/Df) // phase constant in
    degrees
90 p=(%pi/Df)*(d/lambda2)*cosd(b) // numerical distance
91 A=((2+0.3*p)/(2+p+0.6*p^2))-(sind(b)*(sqrt(p/2)*exp
    ((-5*p)/8))) // factor of the ground losses
92 E1=(A*E_not)/(d) // in V/m
93 disp(E1*1000," electric field strength for sea water
    earth in mV/m when f=1500kHz:")
94
95 // for good soil
96 epsilon_r=20 // permittivity
97 sigma=10^-4 // conductivity in mho/cm
98 Df=1.8*10^12*(sigma/f2) // dissipation factor
99 b=atand((epsilon_r+1)/Df) // phase constant in
    degrees
100 p=(%pi/Df)*(d/lambda2)*cosd(b) // numerical distance
101 A=((2+0.3*p)/(2+p+0.6*p^2))-(sind(b)*(sqrt(p/2)*exp
    ((-5*p)/8))) // factor of the ground losses
102 E2=(A*E_not)/(d) // in V/m
103 disp(E2*1000," electric field strength for good soil
    in mV/m when f=1500kHz:")
104
105 // for poor soil
106 epsilon_r=10 // permittivity
107 sigma=0.2*10^-4 // conductivity in mho/cm
108 Df=1.8*10^12*(sigma/f2) // dissipation factor

```

```

109 b=atand((epsilon_r+1)/Df) // phase constant in
    degrees
110 p=(%pi/Df)*(d/lambda2)*cosd(b) // numerical distance
111 A=((2+0.3*p)/(2+p+0.6*p^2))-(sind(b)*(sqrt(p/2)*exp
    ((-5*p)/8))) // factor of the ground losses
112 E3=(A*E_not)/(d) // in V/m
113 disp(E3*1000,"electric field strength for poor soil
    in mV/m when f=1500kHz:")
114
115 // for cities ,industrial areas
116 epsilon_r=5 // permittivity
117 sigma=10^-5 // conductivity in mho/cm
118 Df=1.8*10^12*(sigma/f2) // dissipation factor
119 b=atand((epsilon_r+1)/Df) // phase constant in
    degrees
120 p=(%pi/Df)*(d/lambda2)*cosd(b) // numerical distance
121 A=((2+0.3*p)/(2+p+0.6*p^2))-(sind(b)*(sqrt(p/2)*exp
    ((-5*p)/8))) // factor of the ground losses
122 E4=(A*E_not)/(d) // in V/m
123 disp(E4*1000,"electric field strength for cities ,
    industrial areas in mV/m when f=1500kHz:")
124
125 // for Rocky soil ,flat sandy
126 epsilon_r=10 // permittivity
127 sigma=2*10^-3 // conductivity in mho/cm
128 Df=1.8*10^12*(sigma/f2) // dissipation factor
129 b=atand((epsilon_r+1)/Df) // phase constant in
    degrees
130 p=(%pi/Df)*(d/lambda2)*cosd(b) // numerical distance
131 A=((2+0.3*p)/(2+p+0.6*p^2))-(sind(b)*(sqrt(p/2)*exp
    ((-5*p)/8))) // factor of the ground losses
132 E5=(A*E_not)/(d) // in V/m
133 disp(E5*1000,"electric field strength for Rocky
    soil ,flat sandy in mV/m when f=1500kHz:")
134
135 // for medium hills ,forestation
136 epsilon_r=13 // permittivity
137 sigma=5*10^-5 // conductivity in mho/cm

```

```

138 Df=1.8*10^12*(sigma/f2) // dissipation factor
139 b=atand((epsilon_r+1)/Df) // phase constant in
    degrees
140 p=(%pi/Df)*(d/lambda2)*cosd(b) // numerical distance
141 A=((2+0.3*p)/(2+p+0.6*p^2))-(sind(b)*(sqrt(p/2)*exp
    ((-5*p)/8))) // factor of the ground losses
142 E6=(A*E_not)/(d) // in V/m
143 disp(E6*1000,"electric field strength for medium
    hills ,forestation in mV/m when f=1500kHz:")
144
145
146
147 // notel : misprint value of sigma in part (e) when
    f=500khz correct is sigma=2*10^-3 for rocky soil
148 //note 2: The ans is rounded off at the F = 1500
    kHz in poor soil and Cities , industrial area.

```

---

**Scilab code Exa 9.3** maximum range of tropospheric transmission

```

1 //Exa9.3
2 clc;
3 clear;
4 close;
5 //given :
6 Ht=100 // height of transmittin antenna in feet
7 Hr=50 // height of receiving antenna in feet
8 Dmax=sqrt(2*Ht)+sqrt(2*Hr) // in miles
9 disp(Dmax,"maximum range of tropospheric
    transmission in miles:")

```

---

**Scilab code Exa 9.4** Radio horizon distance

```

1 //Exa 9.4

```

```

2  clc;
3  clear;
4  close;
5  //given :
6  Ht=80 // height of transmitting antenna in meter
7  d=sqrt(17*Ht) // radio horizon distance of
    transmitting antenna
8  disp(d,"radio horizon distance of transmitting
    antenna in Km")

```

---

**Scilab code Exa 9.5** maximum distance covered by the space wave

```

1  //Exa 9.5
2  clc;
3  clear;
4  close;
5  //given :
6  Ht=80 // height of transmitting antenna in meter
7  Hr=50 // height of receiving antenna in meter
8  Dmax=sqrt(17*Ht)+sqrt(17*Hr) // in Km
9  disp(Dmax,"maximum range of tropospheric
    transmission in Km:")

```

---

**Scilab code Exa 9.6** Required height of receiving antenna

```

1  //Exa 9.6
2  clc;
3  clear;
4  close;
5  //given :
6  Dmax=80 //distance in Km
7  Ht=100 // height of transmitting antenna in meter
8  // formula : Dmax=sqrt(17*Ht)+sqrt(17*Hr)

```

```

9 Hr=(Dmax-(sqrt(17*Ht)))^2/17 // height of receiving
  antenna in m
10 disp(Hr,"height of receiving antenna in m")

```

---

**Scilab code Exa 9.7** Radio horizon distance for transmitting and receiving antenna and maximum range

```

1 //Exa 9.7
2 clc;
3 clear;
4 close;
5 // given ;
6 Ht=300 // height of antenna in feet
7 Hr=100 // height of receiving antenna in feet
8 dt=sqrt(2*Ht) // radio horizon distance for a
  transmitting antenna in miles
9 dr=sqrt(2*Hr) // radio horizon distance for a
  transmitting antenna in miles
10 dmax=dt+dr // maximum range of space wave
  propagation in miles
11 disp(dt,"radio horizon distance for a transmitting
  antenna in miles:")
12 disp(dr,"radio horizon distance for a receiving
  antenna in miles:")
13 disp(dmax,"maximum range of space wave propagation
  in miles:")

```

---

**Scilab code Exa 9.8** range of the space wave

```

1 //Exa 9.8
2 clc;
3 clear;
4 close;

```

```

5 // given :
6 f=60 // in MHz
7 f=60*10^6 // in Hertz
8 c=3*10^8 // speed of light in m/s
9 lambda=c/f // wavelength in meter
10 Ptx=1 // transmitting power in kilo watt
11 ht=50 // height of transmitting antenna in meter
12 hr=5 // height of receiving antenna in meter
13 E=80 // electric field in micro V/m
14 E=80*10^(-6) // electric field in V/m
15 E0=3*137.6*sqrt(Ptx)*(8/5)*10^3 // field in mV/m at
    1 meter where Ptx is in kW
16 E0=3*137.6*sqrt(Ptx)*(8/5)*10^3*10^(-3) // field in
    V/m at 1 meter where Ptx is in kW
17 // formula : E=(4*(%pi)*ht*hr*E0)/(lambda*d^2)
18 d=sqrt((4*%pi*ht*hr*E0)/(lambda*E)) // range of
    space wave in meter
19 disp(d/1000,"range of space wave in Km:")

```

---

**Scilab code Exa 9.9** maximum wavelength at which propagation is possible

```

1 //Exa 9.9
2 clc;
3 clear;
4 close;
5 //given :
6 hd=30 // height of duct in m
7 delta_M=30 // unitless
8 LAMBDA_max=2.5*hd*sqrt(delta_M*10^-6) // maximum
    wavelength at which duct propagation is possible
9 disp(LAMBDA_max,"maximum wavelength at which duct
    propagation is possible in m:")

```

---



**Scilab code Exa 9.10** Electron density of the layer

```
1 //Exa 9.10
2 clc;
3 clear;
4 close;
5 // given :
6 fc=1.5 // critical frequency in MHz
7 fc=1.5*10^(6) // critical frequency in Hz
8 // formula : fc=9*sqrt(Nmax)
9 Nmax=(fc)^2/81 // electron density in electrons/m^3
10 disp(Nmax,"electron density in electrons/m^3")
```

---

**Scilab code Exa 9.11** range if the frequency is MUF itself

```
1 // Exa 9.11
2 clc;
3 clear;
4 close;
5 // given :
6 n=0.92 // refractive index
7 MUF=10 // maximum usable frequency in MHz
8 MUF=10*10^6 // maximum usable frequency in Hz
9 f=10*10^6 // in Hz ordinary frequency and maximum
    usable frequency are same
10 h=400 // height of ray reflection point on the
    ionospheric layer in Km
11 h=400*10^3 // height of ray reflection point on the
    ionospheric layer in m
12 // formula :n=sqrt(1-(81*Nmax/f^2))
13 Nmax=(1-n^2)*f^2/81 // electron density in electrons
    /m^3
```

```

14 fc=9*sqrt(Nmax) // critical frequency in Hz
15 // MUF=fc*sec(thetai)
16 //sec(thetai)=MUF/fc
17 // also , sec(thetai)=sqrt(h^2+(d^2/4))/h so on
    comparing ,
18 d=sqrt(((MUF*h/fc)^2-h^2)*4) // range in km
19 disp(d/1000,"Range in Km")
20
21
22 // note :answer in the book is 1876.59 where as in
    scilab is 1877.94 minute difference only

```

---

**Scilab code Exa 9.12** Relative permittivity of D E F layers

```

1 //Exa 9.12
2 clc;
3 clear;
4 close;
5 // given ;
6 f=50 // in hz
7 f=50*10^3 // in KHz
8 N1=400 // electron density of D layer in electrons/
    cm^3
9 N2=5*10^5 // electron density of E layer in
    electrons/cm^3
10 N3=2*10^6 // electron density of F layer in
    electrons/cm^3
11 // formula :n=sqrt(epsilon_r)=sqrt(1-(81*N/f^2)) //
    WHERE f IS IN KhZ
12 // for D layer
13 epsilon_r1=1-(81*N1/f^2) // relative permittivity of
    D layer
14 // for E layer
15 epsilon_r2=1-(81*N2/f^2) // relative permittivity of
    E layer

```

```

16 // for F layer
17 epsilon_r3=1-(81*N3/f^2) // relative permittivity of
    F layer
18 disp(round(epsilon_r1),"relative permittivity of D
    layer:")
19 disp(epsilon_r2,"relative permittivity of E layer:")
20 disp(epsilon_r3,"relative permittivity of F layer:")

```

---

### Scilab code Exa 9.13 Angle of refraction

```

1 // Exa 9.13
2 clc;
3 clear;
4 close;
5 //given :
6 f=50 // in hz
7 f=50*10^3 // in KHz
8 theta_i=30 // in degrees
9 N=400 // electron density of D layer in electrons/cm
    ^3
10 // formula :n=sqrt(epsilon_r)=sqrt(1-(81*N/f^2)) //
    WHERE f IS IN KhZ
11 // for D layer
12 epsilon_r=1-(81*N/f^2) // relative permittivity of D
    layer
13 n=sqrt(epsilon_r) // refractive index
14 // formula :nsin(theta_r)=sin(theta_i) // snell's
    law
15 theta_r=asind(sind(theta_i/n)) // angle of
    refraction in degrees
16 disp(theta_r,"angle of refraction in degrees:")

```

---

### Scilab code Exa 9.14 critical frequency of an electromagnetic wave

```

1 //Exa 9.14
2 clc;
3 clear;
4 close;
5 // given :
6 N1=400 // electron density of D layer in electrons/
        cm^3
7 N2=5*10^5 // electron density of E layer in
        electrons/cm^3
8 N3=2*10^6 // electron density of F layer in
        electrons/cm^3
9 // formula : fc=9*sqrt(N)
10 fc1=9*sqrt(N1) // critical frequency in Khz of EM
        wave for D layer
11 fc2=9*sqrt(N2) // critical frequency in MHz of EM
        wave for E layer
12 fc3=9*sqrt(N3) // critical frequency in MHz of EM
        wave for F layer
13 disp(fc1,"critical frequency of EM wave for D layer
        in kHz:")
14 disp(fc2/10^3,"critical frequency of EM wave for E
        layer in Mhz:")
15 disp(fc3/10^3,"critical frequency of EM wave for F
        layer in Mhz:")
16
17
18
19 //note: the value of fc3 in book is equal to 12.8MHz
        but the correct is 12.72MHz.

```

---

#### Scilab code Exa 9.15 Critical frequency

```

1 // Exa 9.15
2 clc;
3 clear;

```

```

4 close;
5 // given :
6 Nmax=1.3*10^6 // maximum electron density in
   electrons/cm^3
7 // formula : fc=9*sqrt(Nmax)
8 fc_khz=9*sqrt(Nmax) // critical frequency in Khz
9 disp(fc_khz/1000,"critical frequency in Mhz:")

```

---

**Scilab code Exa 9.16** maximum usable frequency

```

1 //Exa 9.16
2 clc;
3 clear;
4 close;
5 // given :
6 d=2600 // distance between the points in Km
7 d=2600*10^3 // distance between the points in m
8 fc=4 // critical frequency in MHz
9 fc=4*10^6 // critical frequency in Hz
10 h=200 // height of ionospheric layer in km
11 h=200*10^3 // height of ionospheric layer in m
12 MUF=fc*sqrt(1+(d/(2*h))^2) // maximum usable
   frequency (this step is Misprinted in the book)
13 disp(MUF/10^6,"maximum usable frequency in MHz:")
14
15
16
17
18 //note: Answer in the book is wrong.

```

---

**Scilab code Exa 9.17** frequency of propagating wave for D layer

```

1 // Exa 9.17

```

```

2  clc;
3  clear;
4  close;
5  // given :
6  N=400 // electron density in electrons/cm^3
7  n=0.5 // refractive index
8  // formula : n=sqrt(1-(81*N/f^2))
9  f=sqrt(81*N/(1-n^2)) // frequency in kHz
10 disp(f,"frequency of propagating wave in kHz:")

```

---

**Scilab code Exa 9.18** Range of line of sight

```

1  //Exa 9.18
2  clc;
3  clear;
4  close;
5  //given :
6  Ht=60 // height of transmitting antenna in meter
7  Hr=6 // height of receiving antenna in meter
8  d=sqrt(17*Ht)+sqrt(17*Hr) // in Km
9  disp(d,"range of line of sight in Km:")

```

---

**Scilab code Exa 9.19** Critical angle of propagation for D layer

```

1  //Exa 9.19
2  clc;
3  clear;
4  close;
5  // given :
6  d=500 // distance between transmitter and receiver
       in km
7  h=70 // height of D layer in km

```

```
8 theta_c=asind(h/(sqrt(h^2+(d^2/4)))) // critical
   angle in degrees
9 disp(theta_c,"critical angle of propagation in
   degrees:")
10
11
12
13 //it can also be calculated from
14 theta_c=atand(2*h/d) // critical angle in degrees
15 disp(theta_c,"critical angle of propagation in
   degrees by method 2:")
```

---