A Study of Dipole Antennas using MatLab®

Robert W. Schuler
Spring 2003
EE 540

Submitted to: Dr. Mark Mirotznik
The Catholic University of America

Executive Summary

This paper is one of five deliverables generated while studying antennas during the Spring 2003 Semester of "EE 540 Introduction to Antenna and Radar" taught at NSWCCD by Dr. Mark Mirotznik of the Catholic University of America. The other four deliverables are in the form of software written to run in MatLab®. The four MatLab algorithms are progressive in nature and follow this sequence:

1. Dipole using analytic equations
2. Dipole using given current distribution
3. Dipole using Method of Moments to calculate current distribution
4. Linear array of Dipole's.

The first algorithm is the simplest in terms of programming complexity; the second algorithm is slightly more complex; the third more complex still; and the fourth algorithm is the most complex. The algorithms also progress in the number of required user input values, with the first needing only three values; the second needing four, the third algorithm asks for five inputs; and the user is asked to provide eight inputs for the array algorithm. Each algorithm calculates and displays the Directivity, Radiation Pattern, Radiated Power, and Radiation Resistance for the given dipole or array.

The material presented in this paper and the accompanying algorithms is based on information from Dr. Mirotznik's lecture notes and on Dr. Constantine A. Balanis' book Antenna Theory: Analysis and Design 2nd edition copyright 1982 by Wiley & Sons, Inc. ISBN 0-471-59268-4. The algorithms where developed using MatLab Version 5.0.1.421 May 25, 1997.

The values calculated by all four algorithms match published values for ½ wavelength dipoles and "infinitesimal" (actually just really short) dipoles. There is some numerical instability evident in the calculation of current using the Method of Moments, which warrants some further study.

Table of Contents

Installation

The four programs delivered with this report consist of 10 individual files:

- ee540_dipole_1.m
- ee540_dipole_2.m
- ee540_dipole_3.m
- ee540_array_4.m
- dipoleint1.m
- dipoleint2.m
- pocklington.m
- calc_current3.m
- array_factor.m
- dipoleint_array.m

Installation consists of simply copying these files into a directory on a computer with MatLab. Once this is done, the user need only start MatLab, change to the directory containing the files listed above, and execute any of the first four files listed.

ee540_dipole_1.m

ee540_dipole_1 – performs an analysis of a dipole antenna centered at the origin and running along the Z-axis. The user is asked to input the wavelength, dipole-length, and maximum current. The program then calculates Directivity, Radiated Power, and Radiation Resistance. The program also graphs Directivity as a function of elevation (theta) and azimuth (phi) angles. The field of Directivity values is also plotted in 3D as a function of both theta and phi.

Execution of ee540_dipole_1

The program is launched by starting MatLab, changing to the directory where the files are installed, and then typing **`ee540_dipole_1`** at the MatLab prompt. The user is then prompted for three pieces of information about the dipole:

1. The wavelength for which the dipole is to operate. The prompt is
   ```
   Wave Length (default=500):
   ```
   By just hitting enter, the default value of 500 is selected. Alternatively, the user may key-in a different value. It is safe to assume meters as the unit, however, the units of this value are not used in the algorithm
2. The antenna length as a fraction of the wavelength. The prompt is
   ```
   Antenna Length as Fraction of the Wavelength
       (.5 for 1/2 wavelength)
       (.005 for an infinitesimal dipole)
       (1.25 to see some side lobes)
   Enter Fraction (default=.5):
   ```
   By just hitting enter, the default value of ½ wavelength is selected. Alternatively, the user may key-in a different value. Two additional values are suggested in the prompt: .005 analyzes a very small (infinitesimal) dipole, and 1.25 (1¼) analyzes a dipole that demonstrates some side-lobes.
3. The maximum current along the dipole ($\mathbf{I_0}$). The prompt is
   ```
   Maximum Current in Amps (default=1):
   ```
   By just hitting enter, the default value of 1 Ampere is selected. Alternatively, the user may key-in a different value. Changing this value only impacts the radiated power (Prad).

The algorithm generates three graphs and reflects the calculated values for Directivity, Power Radiated, and Radiation Resistance.

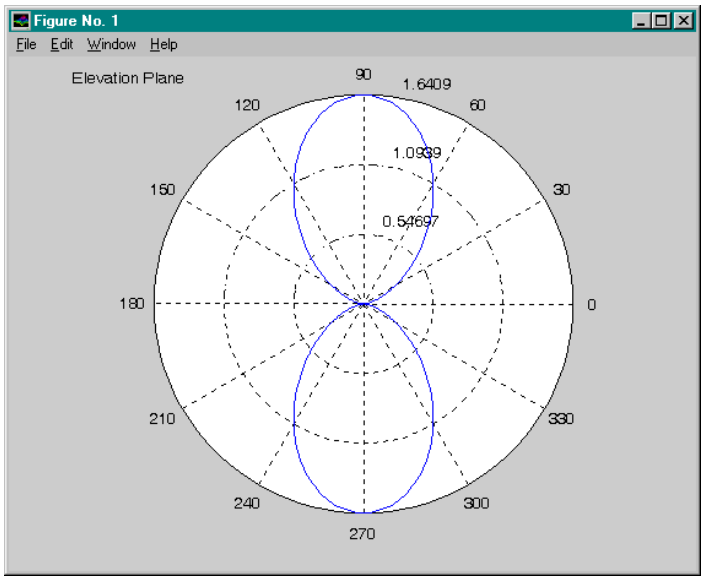


Figure No. 1
File Edit Window Help
Elevation Plane
90
1.6409
120
60
1.0939
150
30
0.54697
180
0
210
330
240
300
270

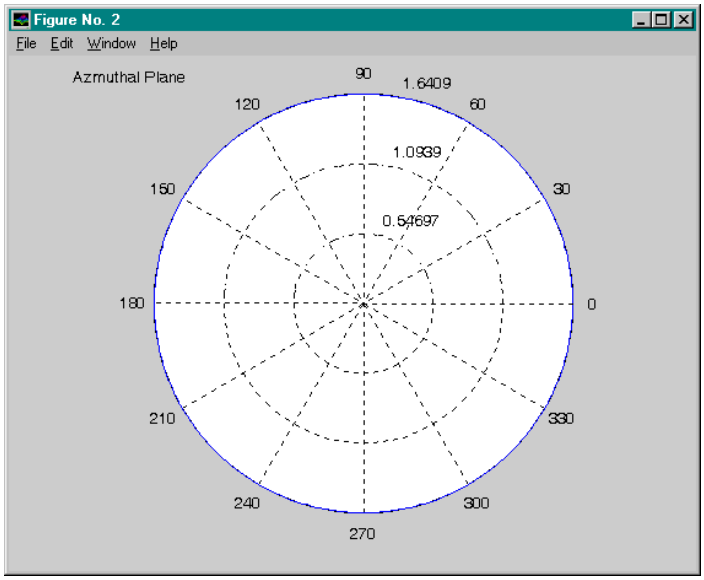
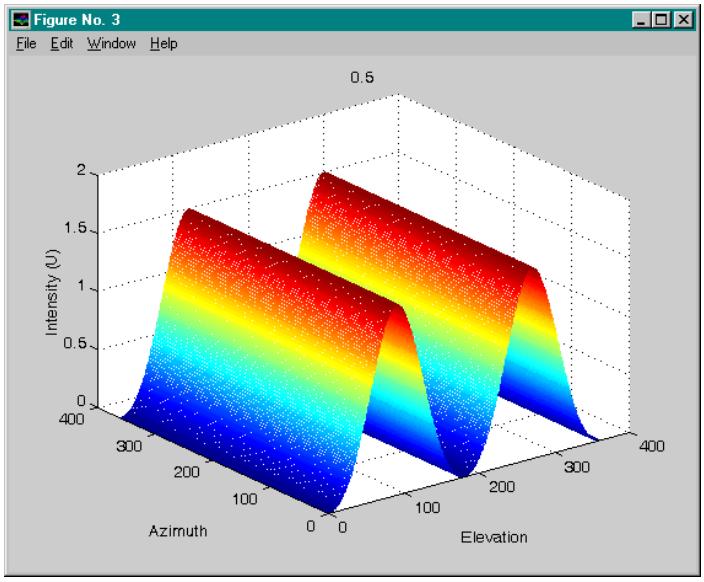
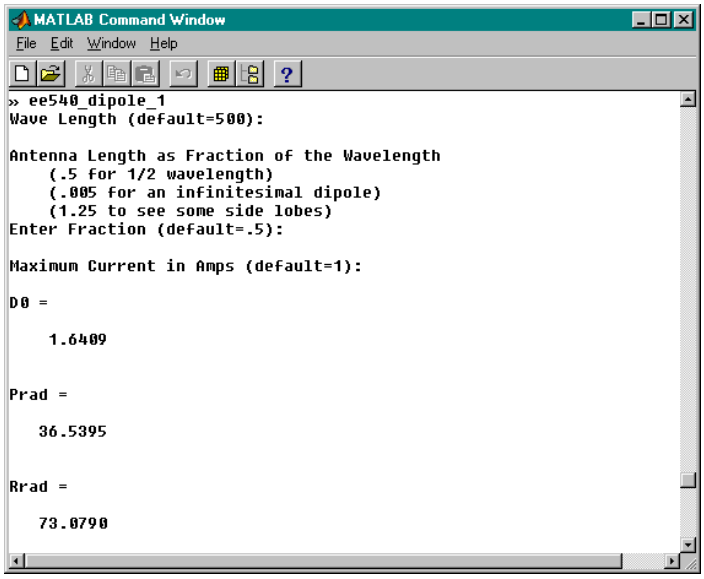


Figure No. 2
File Edit Window Help
Azmuthal Plane
90
1.6409
120
60
1.0939
150
30
0.54697
180
0
210
330
240
300
270





Explanation of ee540_dipole_1

The analysis is done using equation 4-62a on page 153 of Balanis to calculate the E-field ($\mathbf{E_\theta}$) based on a user provided wavelength, dipole-length, and maximum current value. The E-field is then used to calculate the Radiation Intensity (**U**) of the dipole using equation 2-12a on page 38 of Balanis. These two calculations are performed by the code in dipoleint1.m and are invoked from ee540_dipole_1.m by the call:

U = dipoleint1(T, P, lamda, length1, current);

Where T is a vector of values for theta (the elevation angle)
P is a vector of values for phi (the azimuthal angle)
length1 (**l**) is the length of the dipole
current ($\mathbf{I_0}$)is the maximum current along the dipole

The code in dipoleint1 has been isolated from the main body of algorithm in order to modularize the code for future use. In fact the main body of ee540_dipole_1 is reused in ee540_dipole_2 and ee540_dipole_3. In other words, once the Radiation Intensity is known, the same code is used to calculate the total power radiated (Prad), the Directivity (D_0) and the Radiation Resistance (Rrad) in all three dipole algorithms.

Once the radiation intensity (U) is known, the total power radiated (Prad) can be found by integrating over the volume of a sphere according to equation 2-13 on page 38 of Balanis. In the case of an omni directional antenna like a dipole, azimuthal integration reduces to a multiplication by 2π. This fact was utilized to eliminate loops within the MatLab code.

Once the total power radiated (Prad) is known, the Directivity of the antenna can be calculated using equation 2-16 on page 39 of Balanis. This translates into multiplying every value of U by 4π/Prad. The maximum value of the resulting field is the Directivity (D_0) according to equation 2-16a on the same page of Balanis.

Radiation Resistance can be found from Ohm's Law and the Power Law. V=IR and P=IV respectively. R = V/I and V= P/I, so R = P/I^2. This is also shown in equation 4-70 on page 157 of Balanis. Radiation Resistance (Rrad) is equal to Prad divided by the square of the maximum current (I_0). In this case, I_0 is a user input. By varying I_0 Rrad should not change because I_0 is a factor in both the numerator and denominator of the Rrad equation.

ee540_dipole_2.m

ee540_dipole_2 – performs an analysis of a dipole antenna centered at the origin and running along the Z-axis. The user is asked to input the wavelength, dipole-length, maximum current, and the number of constant current segments to assume along the antenna's length. The program then calculates Directivity, Radiated Power, and Radiation Resistance. The program also graphs the calculated current (both magnitude and phase) along the antenna and Directivity as a function of elevation (theta) and azimuth (phi). The field of Directivity values is also plotted in 3D as a function of both theta and phi.

Execution of ee540_dipole_2

The program is launched by starting MatLab, changing to the directory where the files are installed, and then typing **`ee540_dipole_2`** at the MatLab prompt. The user is then prompted for four pieces of information about the dipole:

1. The wavelength for which the dipole is to operate. The prompt is
   ```
   Wave Length (default=500):
   ```
   By just hitting enter, the default value of 500 is selected. Alternatively, the user may key-in a different value. It is safe to assume meters as the unit, however, the units of this value are not used in the algorithm
2. The antenna length as a fraction of the wavelength. The prompt is
   ```
   Antenna Length as Fraction of the Wavelength
        (.5 for 1/2 wavelength)
        (.005 for an infinitesimal dipole)
        (1.25 to see some side lobes)
   Enter Fraction (default=.5):
   ```
   By just hitting enter, the default value of ½ wavelength is selected. Alternatively, the user may key-in a different value. Two additional values are suggested in the prompt: .005 analyzes a very small (infinitesimal) dipole, and 1.25 (1¼) analyzes a dipole that demonstrates some side-lobes.
3. The maximum current along the dipole ($\mathbf{I_0}$). The prompt is
   ```
   Maximum Current in Amps (default=1):
   ```
   By just hitting enter, the default value of 1 Ampere is selected. Alternatively, the user may key-in a different value. Changing this value only impacts the radiated power (Prad).
4. The number of steps (segments of constant current) to assume along the length of the dipole. The prompt is
   ```
   Number of constant current segments to use
    in approximating the current (default=200):
   ```
   By just hitting enter, the default value of 200 is selected. Alternatively, the user may key-in a different value to increase or decrease the granularity of the calculations.

The algorithm generates five graphs and reflects the calculated values for Directivity, Power Radiated, and Radiation Resistance.

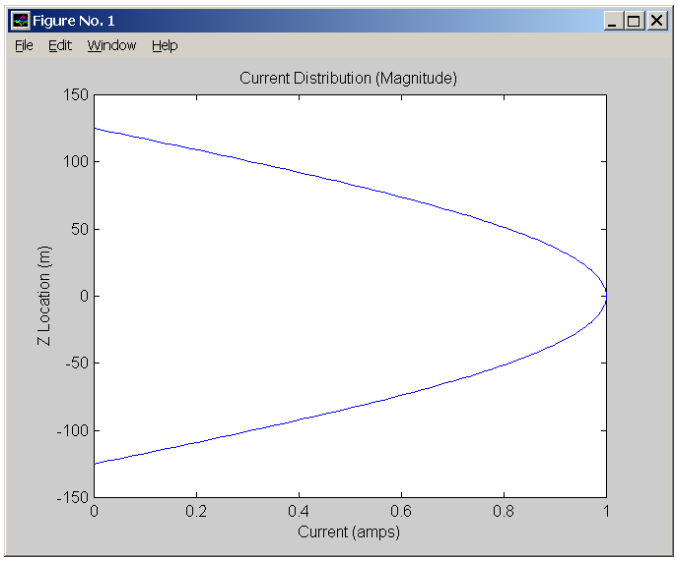


Figure No. 1
File Edit Window Help
Current Distribution (Magnitude)
Z Location (m)
150
100
50
0
-50
-100
-150
0
0.2
0.4
0.6
0.8
1
Current (amps)

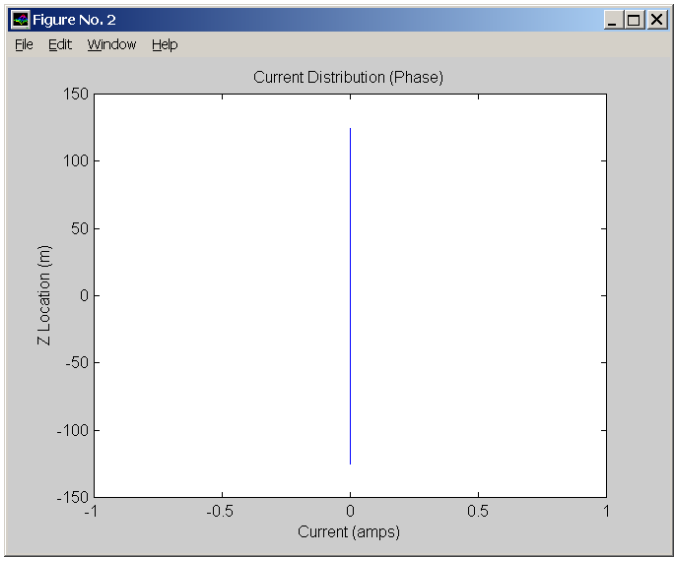


Figure No. 2
File Edit Window Help
Current Distribution (Phase)
Z Location (m)
150
100
50
0
-50
-100
-150
-1
-0.5
0
0.5
1
Current (amps)

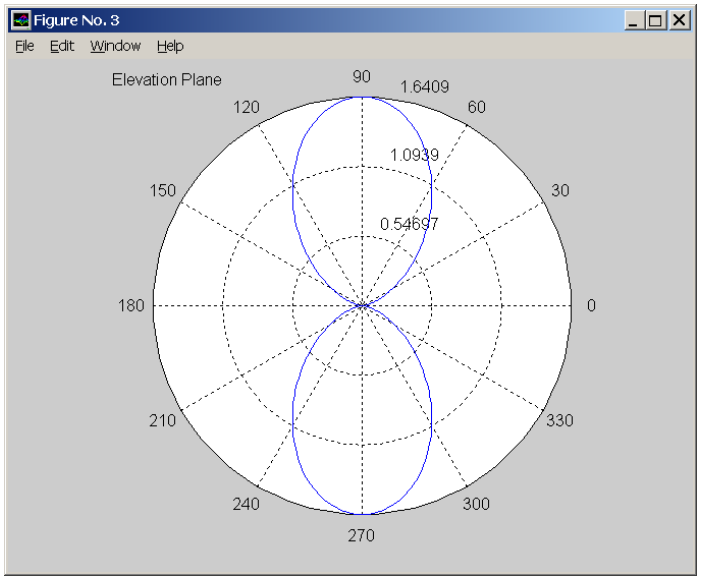


Figure No. 3
File Edit Window Help
Elevation Plane
90
1.6409
120
60
1.0939
150
30
0.54697
180
0
210
330
240
300
270

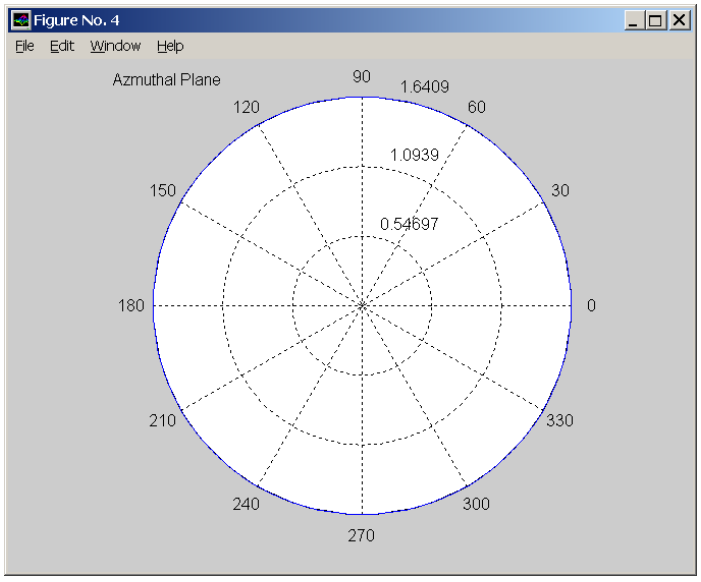
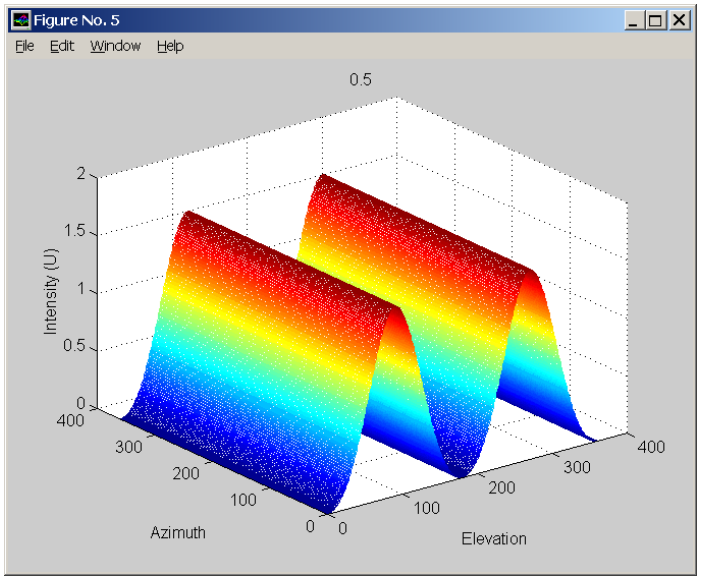
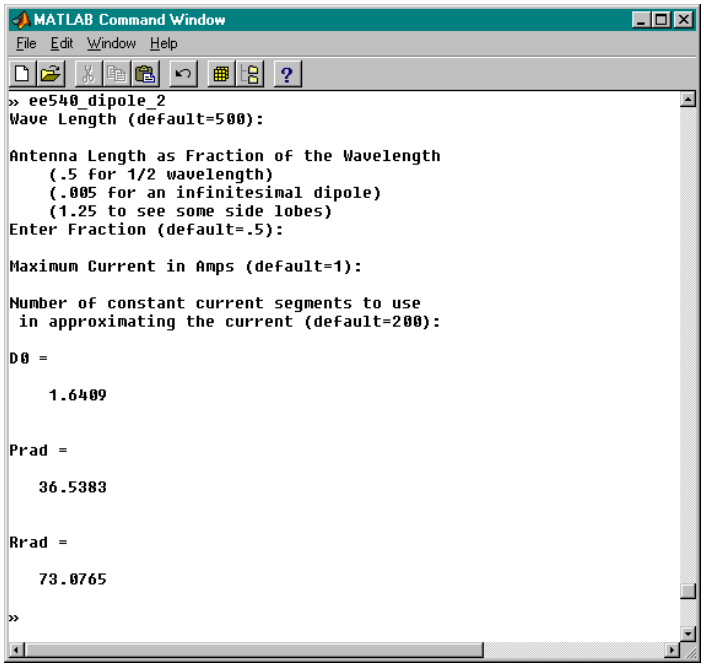


Figure No. 4
File Edit Window Help
Azmuthal Plane
90
1.6409
120
60
1.0939
150
30
0.54697
180
0
210
330
240
300
270





Explanation of ee540_dipole_2

The analysis is done using equation 4-58a on page 152 of Balanis to calculate the E-field ($\mathbf{E_\theta}$) based on a user provided wavelength and dipole-length. In this case, the current distribution along the antenna is calculated from the maximum current and number of constant current segments using equation 4-56 on page 151 of Balanis. It is possible to edit ee540_dipole_2.m to replace ***current*** with a different vector of current values. The actual lines in MatLab are:

```
zprime=linspace(-ant_length/2,ant_length/2,current_steps);
current=max_current*sin(k*(ant_length/2-abs(zprime)));
```

The first line defines an array of z-locations along the dipole.
The second line uses equation 4-56 to calculate a value of current at each location in zprime.

The E-field is then used to calculate the Radiation Intensity (**U**) of the dipole using equation 2-12a on page 38 of Balanis. These two calculations are performed by the code in dipoleint2.m and are invoked from ee540_dipole_2.m by the call:

U = dipoleint2(T, P, lambda, current, zprime);

Where T is a vector of values for theta (the elevation angle)
P is a vector of values for phi (the azimuthal angle)
lambda is the wavelength supplied by the user
current is a vector containing the current for each segment
zprime is a vector containing the z-location of each segment

The code in dipoleint2 has been isolated from the main body of algorithm in order to modularize the code for future use. In fact the main body of ee540_dipole_2 is a direct copy of the code in ee540_dipole_1, with the addition of code to calculate the current distribution. Furthermore, dipoleint2.m is a generic algorithm for calculating Radiation Intensity given a current distribution, and it is reused in ee540_dipole_3.m. In other words, once the current distribution is known, the same code is used to calculate the Radiation Intensity. Once the Radiation Intensity is known, the same code is used to calculate the total power radiated (Prad), the Directivity (D_0) and the Radiation Resistance (Rrad) in all three dipole algorithms.

ee540_dipole_3.m

ee540_dipole_3 – performs an analysis of a dipole antenna centered at the origin and running along the Z-axis. The user is asked to input the wavelength, dipole-length, wire radius, number of constant current segments, and the voltage at the antenna's feed. The program then calculates Directivity, Radiated Power, and Radiation Resistance. The program also graphs the calculated current along the antenna (both magnitude and phase) and Directivity as a function of elevation (theta) and azimuth (phi). The field of Directivity values is also plotted in 3D as a function of both theta and phi.

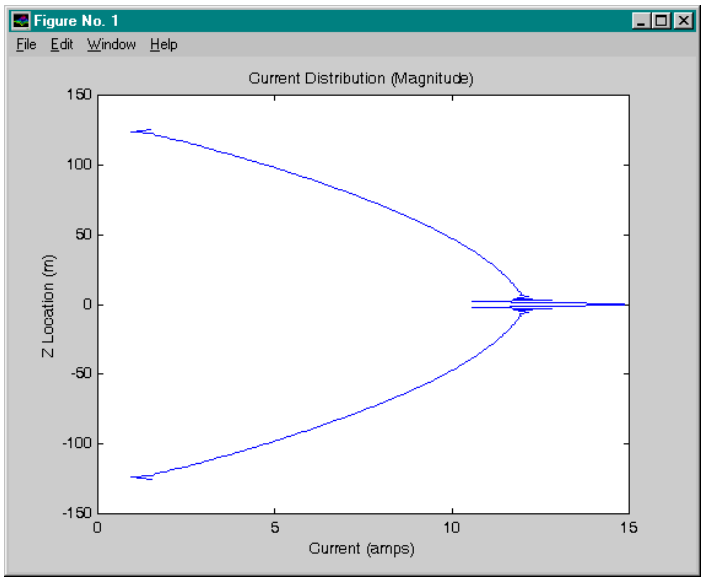
Execution of ee540_dipole_3

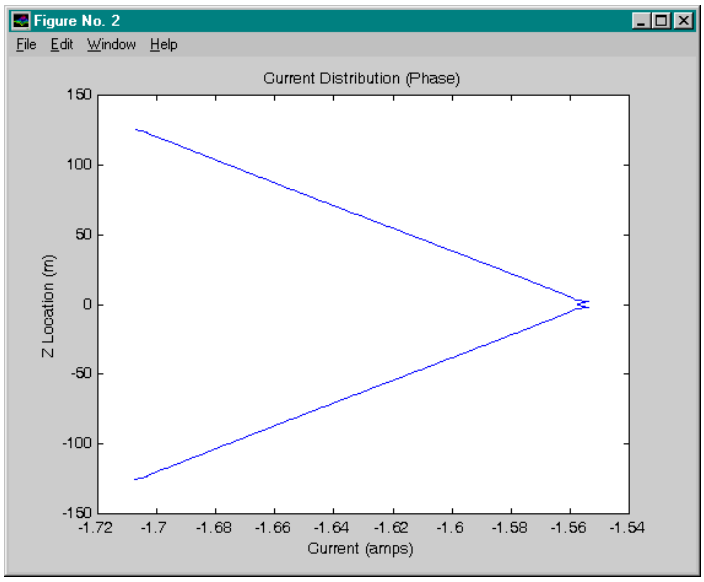
The program is launched by starting MatLab, changing to the directory where the files are installed, and then typing **`ee540_dipole_1`** at the MatLab prompt. The user is then prompted for five pieces of information about the dipole:

1. The wavelength for which the dipole is to operate. The prompt is
   ```
   Wave Length (default=500):
   ```
   By just hitting enter, the default value of 500 is selected. Alternatively, the user may key-in a different value. It is safe to assume meters as the unit, however, the units of this value are not used in the algorithm
2. The antenna length as a fraction of the wavelength. The prompt is
   ```
   Antenna Length as Fraction of the Wavelength
       (.5 for 1/2 wavelength)
       (.005 for an infinitesimal dipole)
       (1.25 to see some side lobes)
   Enter Fraction (default=.5):
   ```
   By just hitting enter, the default value of ½ wavelength is selected. Alternatively, the user may key-in a different value. Two additional values are suggested in the prompt: .005 analyzes a very small (infinitesimal) dipole, and 1.25 (1¼) analyzes a dipole that demonstrates some side-lobes.
3. The antenna radius as a fraction of wavelength. The prompt is
   ```
   Antenna Radius as a fraction of Wavelength
     (.005 for thin wire)
     1/100 of dipole lenght is recommended
   Enter Radius Fraction (default=.005):
   ```
4. The number of steps (segments of constant current) to assume along the length of the dipole. The prompt is
   ```
   Number of constant current segments to use
    in approximating the current
    (must be an odd number: default=199):
   ```
   By just hitting enter, the default value of 200 is selected. Alternatively, the user may key-in a different value to increase or decrease the granularity of the calculations.
5. The voltage at the antenna's feed point. The prompt is
   ```
   Peak Voltage at Feed (default=.01*j):
   ```
   By just hitting enter, the default value of 0+0.1j is selected. Alternatively,

the user may key-in a different value. Changing this value only impacts the radiated power (Prad).

The algorithm generates five graphs and reflects the calculated values for Directivity, Power Radiated, and Radiation Resistance.





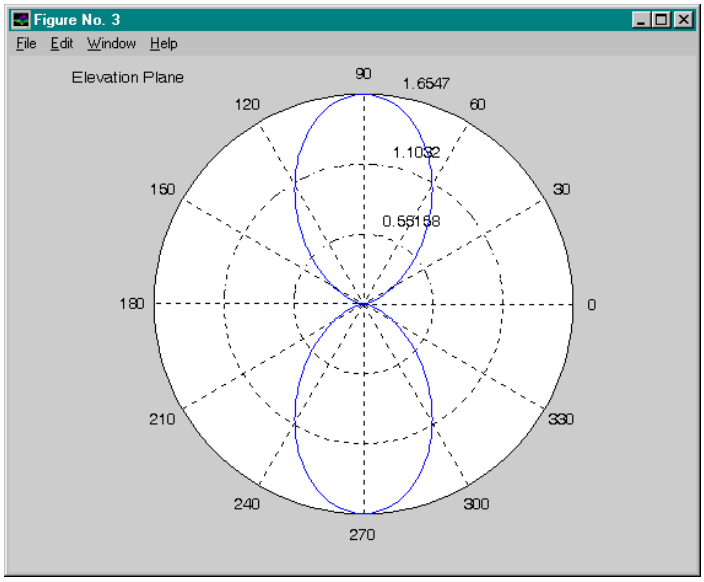


Figure No. 3
File Edit Window Help
Elevation Plane
90
1.6547
120
60
1.1032
150
30
0.55158
180
0
210
330
240
300
270

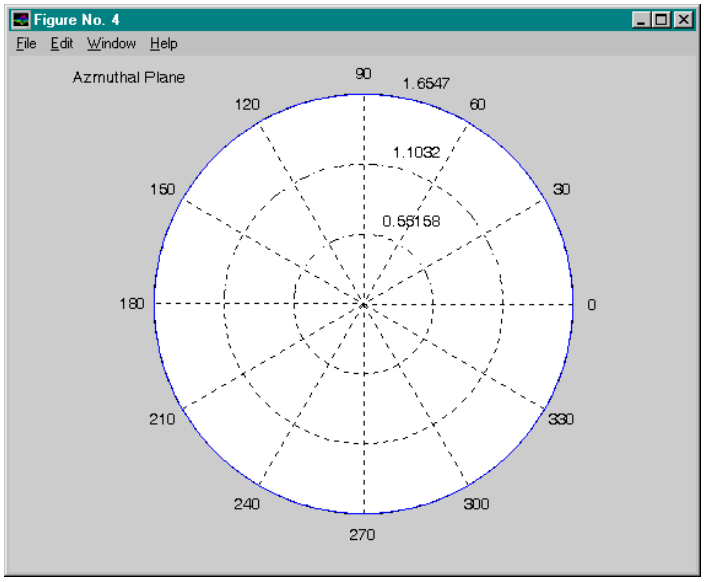
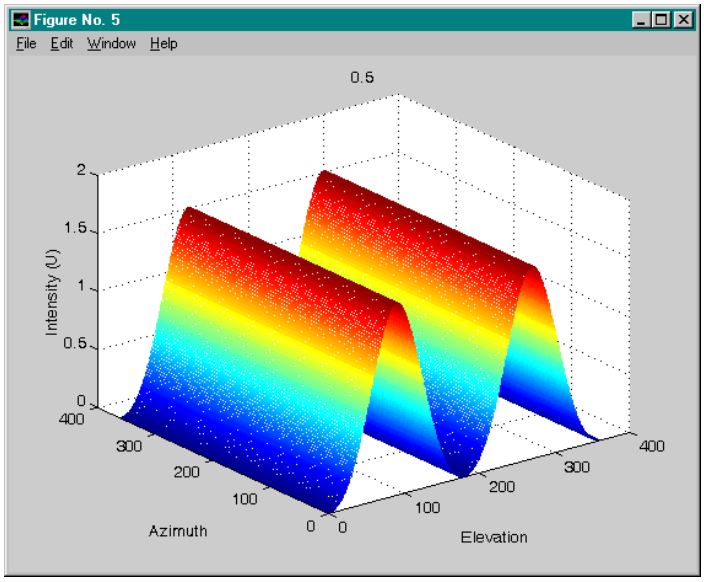
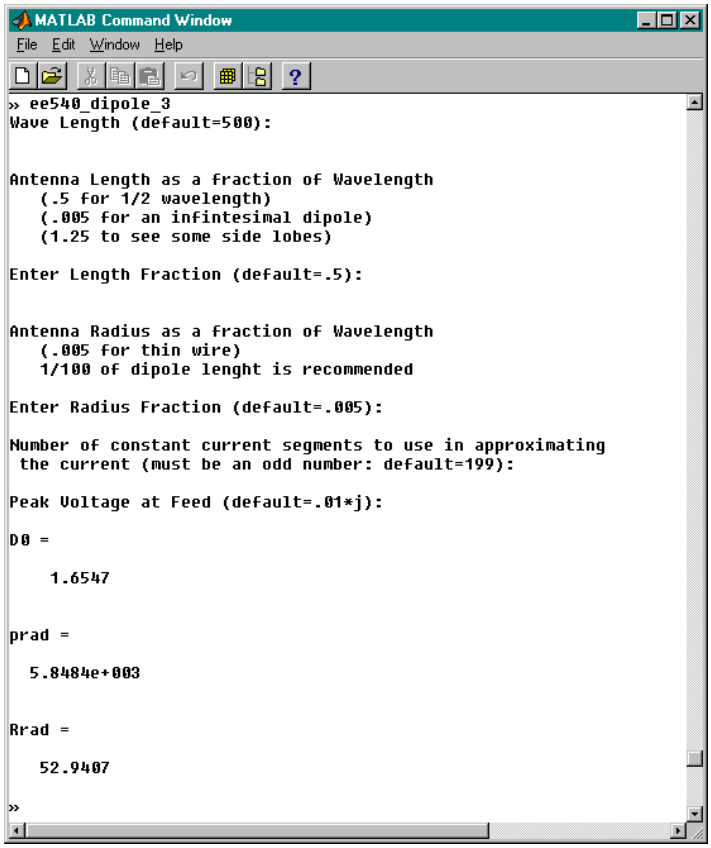


Figure No. 4
File Edit Window Help
Azmuthal Plane
90
1.6547
120
60
1.1032
150
30
0.55158
180
0
210
330
240
300
270





Explanation of ee540_dipole_3

Just like ee540_dipole_2, the analysis is done using equation 4-62a on page 153 of Balanis to calculate the E-field ($\mathbf{E_\theta}$) based on a user provided wavelength and dipole-length. In this case, however, the current distribution is found by using the Method of Moments using Pocklington's Integral equation and point matching. In addition to the

information required in ee540_dipole_2, this algorithm also needs the radius of the wire making up the dipole as well as the feed voltage at the terminal of the dipole. The same equation is used for calculating the z-positions along the dipole:

```
zprime=linspace(-ant_length/2,ant_length/2,current_steps);
```

In this case, however, the calculation of current is much more involved, and the code has been put into a separate file (**calc_current3.m**), which is invoked with the following call:

```
calc_current3(lambda,ant_length,wire_radius,zprime,feed_voltage);
```

Where lambda is the user supplied wavelength
ant_length is the physical length of the dipole
wire_radius is the physical radius of the dipole
zprime is the vector of points along the dipole
feed_voltage is the voltage input at the center of the dipole

This routine uses Pocklington's Integral Equation to define the top row of a matrix used to calculate the current along the dipole given the voltage:

$$\mathbf{Z}*\mathbf{I} = \mathbf{V}$$

Z possesses the Toeplitz property, which means that after computing the first row, the entire matrix can be filled-in using the MatLab TOEPLITZ function. The resulting matrix has the form:

```
|A B C D E F G . . . Z|
|B A B C D E F . . . Y|
|c B A B C D E . . . X|
|.             .     .|
|.               .   .|
|.                 . .|
|Z Y X W V U T . . . A|
```

V is all zeros except for the center entry. The center entry holds the feed voltage and represents the gap between the two dipole halves.

I is then found to be **V**/**Z**.

Once the current has been calculated, the rest of ee540_dipole_3 is identical to ee540_dipole_2. The calculations of D_0, Prad, and Rrad are the same as those in ee540_dipole_2 and ee540_dipole_1.

ee540_array.m

ee540_array – performs an analysis of a linear array of two or more dipole antennas. The array is centered at the origin and each dipole is parallel to the Z-axis. The user is asked to input the number of antennas, the phase difference between successive antennas, wavelength, antenna spacing, dipole-length, wire radius, number of constant current segments, and the voltage at the antenna's feed. The program then calculates an array factor, the current distribution along an individual dipole, the total Directivity, the total Radiated Power, and the Radiation Resistance of the array. The program also graphs the array factor, the calculated current along the antenna (both magnitude and phase) and Directivity as a function of elevation (theta) and azimuth (phi). The field of Directivity values is also plotted in 3D as a function of both theta and phi.

Execution of ee540_array

The program is launched by starting MatLab, changing to the directory where the files are installed, and then typing **`ee540_array`** at the MatLab prompt. The user is then prompted for eight pieces of information about the dipole:

1. The number of antennas in the array. The prompt is
   ```
   Number of dipoles in this
   linear array (default=7):
   ```
   By just hitting enter, the default value of 7 is selected. Alternatively, the user may key-in a different number of antennas. Selecting 1 antenna produces the same results as ee540_dipole_3.m.
2. The phase shift between successive array elements. The prompt is
   ```
   Phase by which each dipole`s current
   leads its predecessor (default=0 radians):
   ```
   By just hitting enter, the default value of zero radians is selected. Alternatively, the user may key-in a different phase shift. NOTE: Since MatLab "knows" about π the user may enter pi/2 at this prompt for a 90-degree phase shift.
3. The wavelength for which the dipole is to operate. The prompt is
   ```
   Wave Length (default=500):
   ```
   By just hitting enter, the default value of 500 is selected. Alternatively, the user may key-in a different value. It is safe to assume meters as the unit, however, the units of this value are not used in the algorithm
4. The distance between successive antennas in the array. The prompt is
   ```
   Distance between dipoles as a
   multiple of Wavelength (default=2):
   ```
   By just hitting enter, the default value of 2*lambda is selected. Alternatively, the user may key-in a different mutlitple.
5. The antenna length as a fraction of the wavelength. The prompt is
   ```
   Antenna Length as Fraction of the Wavelength
       (.5 for 1/2 wavelength)
       (.005 for an infinitesimal dipole)
       (1.25 to see some side lobes)
   Enter Fraction (default=.5):
   ```

By just hitting enter, the default value of ½ wavelength is selected. Alternatively, the user may key-in a different value. Two additional values are suggested in the prompt: .005 analyzes a very small (infinitesimal) dipole, and 1.25 (1¼) analyzes a dipole that demonstrates some side-lobes.

6. The antenna radius as a fraction of wavelength. The prompt is

```
Antenna Radius as a fraction of Wavelength
 (.005 for thin wire)
 1/100 of dipole lenght is recommended
Enter Radius Fraction (default=.005):
```

7. The number of steps (segments of constant current) to assume along the length of the dipole. The prompt is

```
Number of constant current segments to use
 in approximating the current
 (must be an odd number: default=199):
```

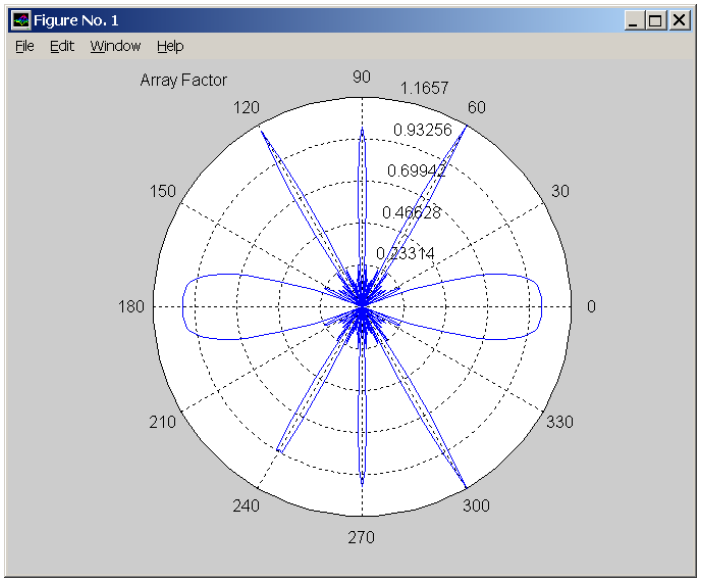
By just hitting enter, the default value of 200 is selected. Alternatively, the user may key-in a different value to increase or decrease the granularity of the calculations.

8. The voltage at the antenna's feed point. The prompt is

```
Peak Voltage at Feed (default=.01*j):
```

By just hitting enter, the default value of 0+0.1j is selected. Alternatively, the user may key-in a different value. Changing this value only impacts the radiated power (Prad).

The algorithm generates six graphs and reflects the calculated values for Directivity, Power Radiated, and Radiation Resistance.



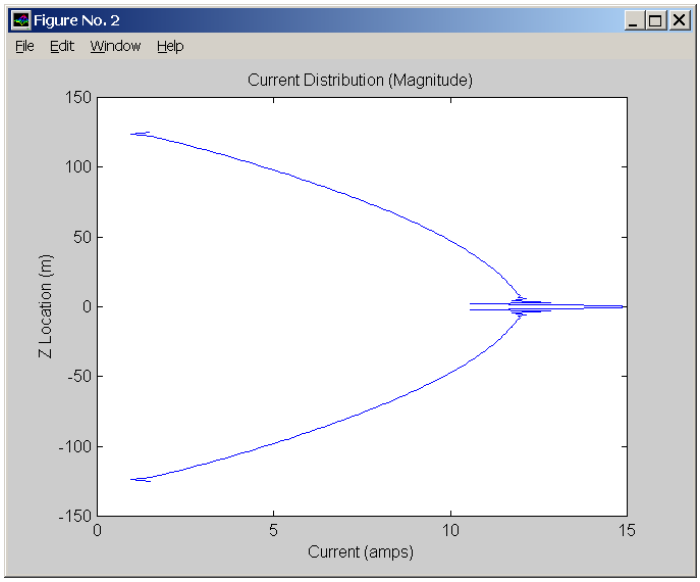


Figure No. 2
File Edit Window Help
Current Distribution (Magnitude)
Z Location (m)
150
100
50
0
-50
-100
-150
0
5
10
15
Current (amps)

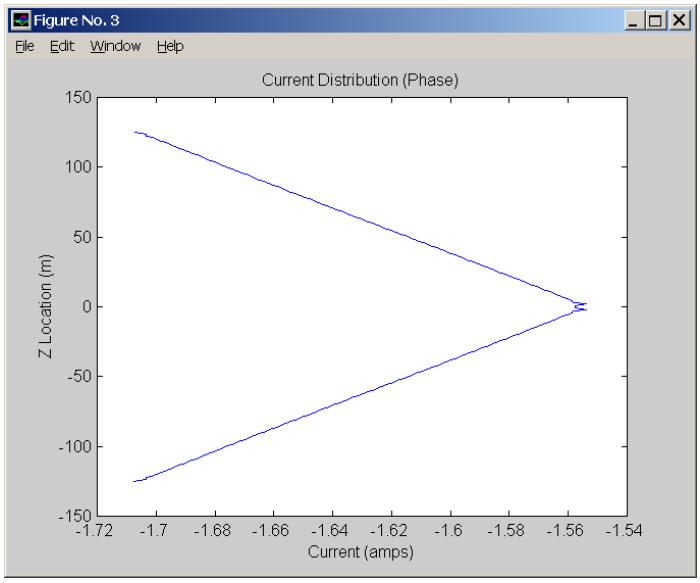


Figure No. 3
File Edit Window Help
Current Distribution (Phase)
Z Location (m)
150
100
50
0
-50
-100
-150
-1.72
-1.7
-1.68
-1.66
-1.64
-1.62
-1.6
-1.58
-1.56
-1.54
Current (amps)

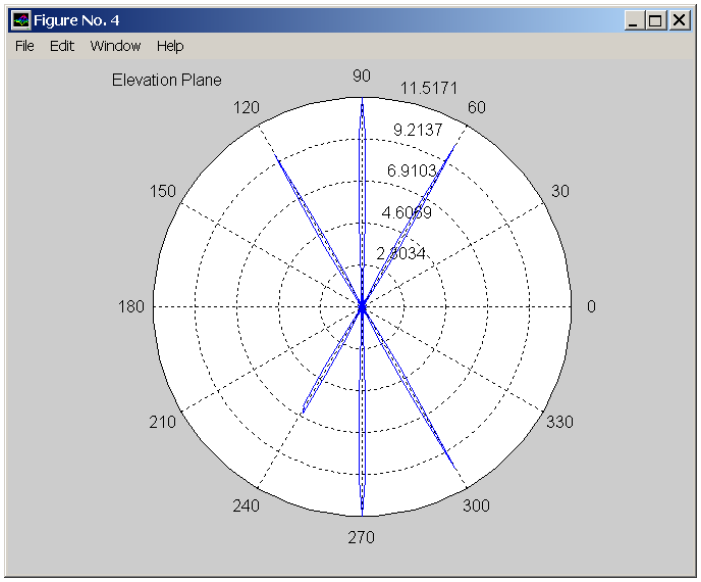


Figure No. 4
File Edit Window Help
Elevation Plane
90
11.5171
120
60
9.2137
6.9103
150
30
4.6069
2.3034
180
0
210
330
240
300
270

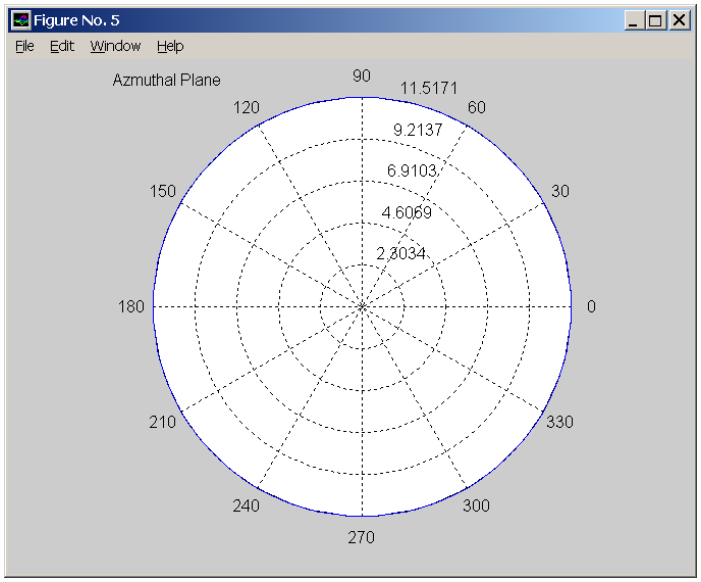


Figure No. 5
File Edit Window Help
Azmuthal Plane
90
11.5171
120
60
9.2137
6.9103
150
30
4.6069
2.3034
180
0
210
330
240
300
270

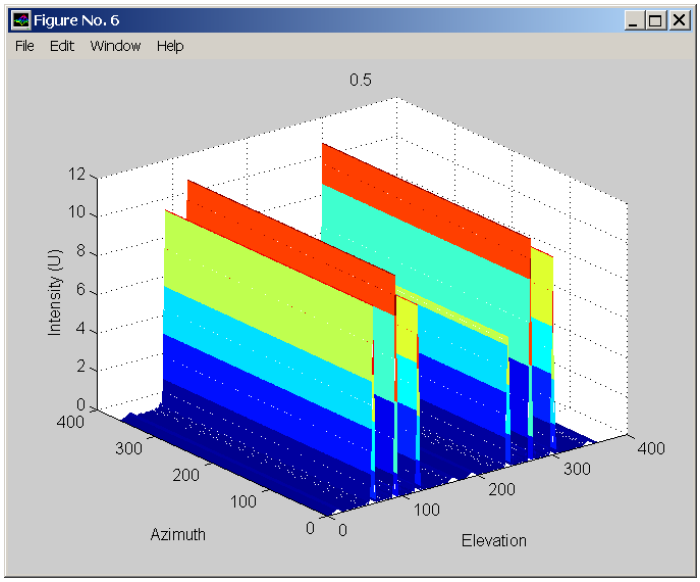
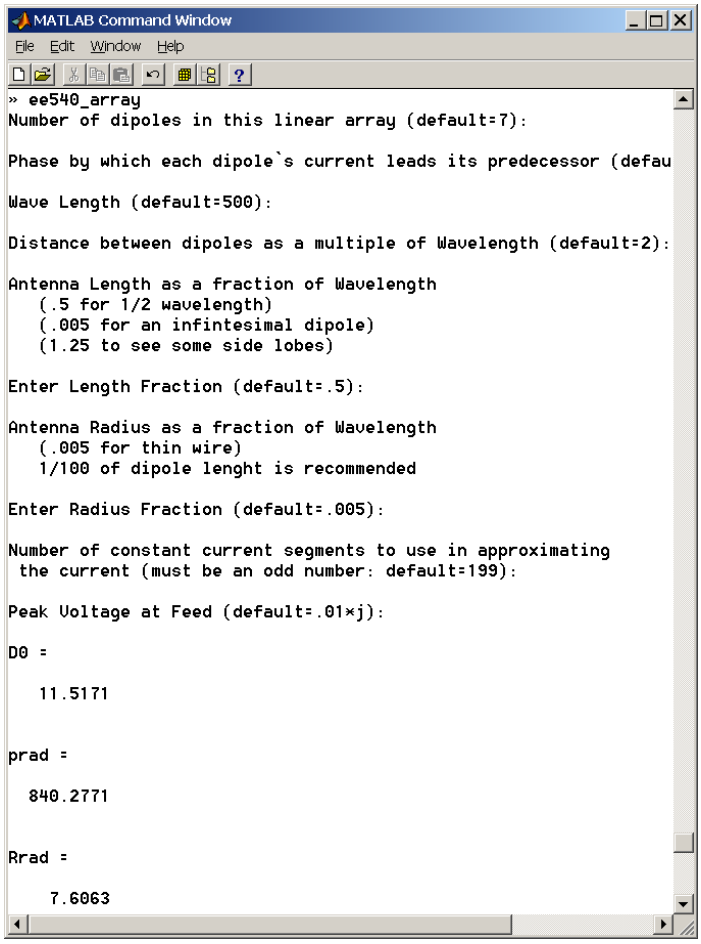


Figure No. 6
File Edit Window Help
0.5
12
10
8
6
4
2
0
Intensity (U)
400
300
200
100
0
Azimuth
0
100
200
300
400
Elevation


MATLAB Command Window
File Edit Window Help
» ee540_array
Number of dipoles in this linear array (default=7):
Phase by which each dipole`s current leads its predecessor (defau
Wave Length (default=500):
Distance between dipoles as a multiple of Wavelength (default=2):
Antenna Length as a fraction of Wavelength
(.5 for 1/2 wavelength)
(.005 for an infintesimal dipole)
(1.25 to see some side lobes)
Enter Length Fraction (default=.5):
Antenna Radius as a fraction of Wavelength
(.005 for thin wire)
1/100 of dipole lenght is recommended
Enter Radius Fraction (default=.005):
Number of constant current segments to use in approximating
the current (must be an odd number: default=199):
Peak Voltage at Feed (default=.01×j):
D0 =
11.5171
prad =
840.2771
Rrad =
7.6063

Explanation of ee540_array

This algorithm extends ee540_dipole_3 by adding an array factor, which is calculated using equation 6-10a on page 259 of Balanis. The analysis of a single dipole is still done using equation 4-62a on page 153 of Balanis to calculate the E-field ($\mathbf{E_\theta}$) based on a user provided wavelength and dipole-length. In this case, however, the E-field is multiplied by the array factor before calculating the Radiation Intensity.

Just like ee540_dipole_3, the current distribution is found by using the Method of Moments using Pocklington's Integral equation and point matching. This is why the user is prompted for the radius of the wire making up the dipole as well as the feed voltage at the terminal of the dipole.

The same code written for ee540_dipole_3 is used to calculate the current along a single dipole (**calc_current3.m**), and is invoked with the following call:

```
calc_current3(lambda,ant_length,wire_radius,zprime,feed_voltage);
```

Where lambda is the user supplied wavelength
ant_length is the physical length of the dipole
wire_radius is the physical radius of the dipole
zprime is the vector of points along the dipole
feed_voltage is the voltage input at the center of the dipole

To accommodate the Array Factor, a new version of the code to calculate the Radiation Intensity had to be written. dipoleint_array.m is an extension of dipoleint3.m, which accounts for the array factor. It is invoked with the following call:

```
U = dipoleint_array(T, P, lambda, current, zprime, AF);
```

Where T is an array of theta values (θ)
P is an array of phi values (φ)
lambda is the user supplied wavelength
current contains the values of current for each antenna segment
zprime is the vector of points along the dipole
AF is the array factor

Once the radiation intensity has been calculated, the rest of ee540_array is identical to ee540_dipole_3. The calculations of D_0, and Prad are the same as those in ee540_dipole_3, ee540_dipole_2, and ee540_dipole_1. The calculation of Rrad is the radiation resistance for the entire array. Rrad is found by dividing the total power radiated by the current assumed for a single antenna times the number of elements in the array. Recall that parallel currents are additive.

Comparison Half-Wavelength

The following table shows results of all four algorithms run with a half-wavelength dipole at 500m wavelength.

	D0	Prad	Rrad
ee540_dipole_1 lambda=500 length=0.5 current=1A	1.6409	36.5395	73.0790
ee540_dipole_2 lambda=500 length=0.5 current=1A segments=200	1.6409	36.5383	73.0765
ee540_dipole_3 lambda=500 length=0.5 radius=0.005 voltage=j0.01 segments=200	1.6593	5.8322e+003	52.7940
ee540_array elements=1 beta=0 lambda=500 distance=0 length=0.5 radius=0.005 voltage=j0.01 segments=200	1.6593	5.8322e+003	52.7940

From published works (and class notes) it is known that the Directivity of a ½ wavelength antenna should be 1.64. The deviation in the second decimal place for ee540_dipole_3 and ee540_array is attributed to the "extra-waviness" of the current being returned from calc_current3.m.

From published works (and class notes) it is known that the Radiation Resistance of a ½ wavelength dipole is close to 75 ohms. Again, I suspect the error in calc_current3.m is contributing to the severe deviation for ee540_dipole_3 and ee540_array.

Prad is dependant on the power input to the antenna. In ee540_dipole_1 and ee540_dipole_2, this current is directly input. In ee540_dipole_3 and ee540_array, the current is calculated from the voltage, which is input. It is inevitable that these values will deviate. Given enough time, it is possible to calculate an input voltage that will result in a maximum current of 1 amp, but this has not been done.

Comparison Very Short Dipole

The following table shows results of all four algorithms run with a very small dipole at 500m wavelength.

	D0	Prad	Rrad
ee540_dipole_1 lambda=500 length=0.005 current=1A	1.5000	6.0836e-007	1.2167e-006
ee540_dipole_2 lambda=500 length=0.005 current=1A segments=200	1.5000	6.0832e-007	1.2166e-006
ee540_dipole_3 lambda=500 length=0.005 radius=0.00005 voltage=j0.01 segments=200	1.5000	4.0128e-011	4.7425e-004
ee540_array elements=1 beta=0 lambda=500 distance=0 length=0.005 radius=0.00005 voltage=j0.01 segments=200	1.5000	4.0128e-011	4.7425e-004

From published works (and class notes) it is known that the Directivity of a infinitesimal dipole antenna should be 1.5. All four algorithms agree on this value.

From published works (and class notes) it is known that the Radiation Resistance of an infinitesimal dipole is very small. I suspect the error in calc_current3.m is contributing to the deviation between the first two algorithms and the second two algorithms.

Prad is dependant on the power input to the antenna. In ee540_dipole_1 and ee540_dipole_2, this current is directly input. In ee540_dipole_3 and ee540_array, the current is calculated from the voltage, which is input. It is inevitable that these values will deviate. Given enough time, it is possible to calculate an input voltage that will result in a maximum current of 1 amp, but this has not been done.

Comparison One and a Quarter-Wavelength

The following table shows results of all four algorithms run with a 1.25*wavelength dipole at 500m wavelength.

	D0	Prad	Rrad
ee540_dipole_1 lambda=500 length=1.25 current=1A	3.2825	53.2316	106.4632
ee540_dipole_2 lambda=500 length=1.25 current=1A segments=200	3.2824	53.2277	106.4554
ee540_dipole_3 lambda=500 length=1.25 radius=0.005 voltage=j0.01 segments=200	3.3117	496.8645	103.0728
ee540_array elements=1 beta=0 lambda=500 distance=0 length=1.25 radius=0.005 voltage=j0.01 segments=200	3.3117	496.8645	103.0728

I suspect the error in calc_current3.m is contributing to the deviation between the first two algorithms and the second two algorithms.

Conclusions

Using MatLab to explore the equations governing the physics of antennas has been very instructional. I wish I had more time to explore the extra "ringing" in the current being calculated by calc_current3.m because I think this is fundamental to the deviations between the outputs of the four algorithms. It would also be instructive to compare these results with the results of the FORTRAN codes provided with Balanis' book.

The algorithms presented in this paper are producing results that match what is known from the book and class notes. There are some deviations which seem due to numerical problems in calculating current. Further study is required to fully explain the deviations.

File Listings

ee540_dipole_1.m

```
function ee540_dipole_1
%ee540_dipole_1 --
%       calculate the Directivity (D0), Radiation Resistance (Rrad),
%       and Radiated Power (Prad)
%       for a given wavelength (lamda), dipole length, and max current.
%
%       These calculations are for a dipole antenna of any length,
%       centered at the origin and running along the z-axis.
%

lamda = input('Wave Length (default=500): ');
if isempty(lamda)
   lamda = 500;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%default value for lamda = 500
end

multiplier = input('\nAntenna Length as Fraction of the Wavelength\n    (.5 for 1/2
wavelength)\n    (.005 for an infinitesimal dipole)\n    (1.25 to see some side
lobes)\nEnter Fraction (default=.5): ');
if isempty(multiplier)
   multiplier = .5;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%default value for mutliplier = 0.5
end
ant_length= multiplier*lamda;  %calculate the length of the antenna

current = input('\nMaximum Current in Amps (default=1): ');
if isempty(current)
   current=1; %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%default value for current = 1
end



Prad = 0;

T=[0:2*pi/360:2*pi];
P=[0:2*pi/360:2*pi];

U = dipoleint1(T, P, lamda, ant_length, current);

%Calculate the total power radiated by "integrating" all values of the
% radiation intensity times sin(theta)*dTheta*dphi
% from theta = 0 to theta = pi and from phi=0 to 2*pi;
% The next 4 lines implement equation 2-13 on page 38 of Balanis
semicircle1= sin(T(1:181));                %queue up sin(theta)
prad1 = U(181,1:181);                      %pluck out one planar slice
prad1 = prad1.*(pi/180).*semicircle1;      %multiply by sin(theta)*dTheta
Prad = sum(prad1)*2*pi;                    %integrate by summing and then multiply
                                           % by dPhi from 0 to 2*pi. This assumes an
                                           % omnidirectional pattern.

%same Prad calculation using loops for the double integration.
%Prad=0;
%for (i=1:361)
%   for (j=1:181)
%      Prad = Prad + U(i,j)*sin(T(j))*(pi/180)^2;
%   end
%end

DM=4*pi*U./Prad; %DM a field of Directivity Magnitudes normalized according to
                 % equation 2-16 on page 39 of Balanis.

D0 = max(max(DM)); % Directivity is the maximum value in this field.

%Elevation vs. Directivity
M1=DM(91,:); %pluck out the graph that sweeps theta while phi=90
figure; polar(T,M1);title('Elevation Plane
');
```

```
%Azimuth vs. Directivity
M2=DM(:,91); %pluck out the graph that sweeps  phi  while theta=90
figure; polar(P,M2');title('Azmuthal Plane
');

%Plot the whole Directivity Magnitude field.
figure; mesh(DM);
xlabel('Elevation');
ylabel('Azimuth');
zlabel('Intensity (U)');
title(ant_length/lamda);

Rrad = 2*Prad/current^2;

%Reflect the three desired parameters back to the user:
D0
Prad
Rrad
```

ee540_dipole_2.m

```
function ee540_dipole_2
%ee540_dipole_2 --
%        Calculate the Directivity (D0), Radiation Resistance (Rrad), and
%        Radiated Power (Prad) for a given wavelength (lamda) and dipole length.
%
%        This routine is very similar to ee540_dipole_1.m
%        In this version of the altorithm, the current distribution is an input
%        to the Radiation Intensity "sub-routine" dipoleint2.m. A default current
%        distribution is calculated from a given maximum current and number of
%        current segments (current steps) along the dipole using equation
%        4-58a on page 152 of Balanis.
%
%        These calculations are for a dipole antenna of any length,
%        centered at the origin and running along the z-axis.
%


% Prompt the User for lamda, antenna length (in terms of lamda), and peak current
lamda = input('Wave Length (default=500): ');
if isempty(lamda)
   lamda = 500;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%default value for lamda = 500
end

multiplier = input('\nAntenna Length as Fraction of the Wavelength\n    (.5 for 1/2
wavelength)\n    (.005 for an infinitesimal dipole)\n    (1.25 to see some side
lobes)\nEnter Fraction (default=.5): ');
if isempty(multiplier)
   multiplier = .5;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%default value for mutliplier = 0.5
end
ant_length = multiplier*lamda;  %calculate the length of the antenna

max_current = input('\nMaximum Current in Amps (default=1): ');
if isempty(max_current)
   max_current=1; %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%default value for current = 1
end

current_steps = input('\nNumber of constant current segments to use\n in approximating
the current (default=200): ');
if isempty(current_steps)
   current_steps=200; %%%%%%%%%%%%%%%%%%%%%%%%%%%%%default value for current_steps = 200
end

%k is the wave number (called beta in some contexts).
k=2*pi/lamda;

%zprime holds the z-locations for the end of each current segment.
zprime=linspace(-ant_length/2,ant_length/2,current_steps);

%current holds an approximation of the current along each segment of the antenna.
%this code uses equation 4-56 on page 151 of Balanis.
%you may replace this array with your own approximation of current.
current=max_current*sin(k*(ant_length/2-abs(zprime)));

%plot the current distribution along the antenna
figure; plot(abs(current),zprime);  title('Current Distribution (Magnitude)');
xlabel('Current (amps)'); ylabel('Z Location (m)');

%plot the current distribution along the antenna
figure; plot(angle(current),zprime);  title('Current Distribution (Phase)');
xlabel('Current (amps)'); ylabel('Z Location (m)');

Prad = 0;

T=[0:2*pi/360:2*pi];
P=[0:2*pi/360:2*pi];

U = dipoleint2(T, P, lamda, current, zprime);
```

```
%Calculate the total power radiated by "integrating" all values of the
% radiation intensity times sin(theta)*dTheta*dphi
% from theta = 0 to theta = pi;
% from phi=0 to 2*pi;
semicircle = sin(T(1:181)); %semicircle is a vector of sin(theta) from 0 to pi
prad1 = U(181,1:181); %at this point, prad1 sweeps through the Radiation Intensity Field
from theta = 0:pi
prad1 = prad1.*(pi/180).*semicircle; %now Prad is U*sin(theta)*dtheta
Prad = sum(prad1)*2*pi; %Prad is the "area" under prad1 rotated 2*pi around phi.

%same Prad calculation using loops for the double integration.
%Prad=0;
%for (i=1:361)
%   for (j=1:181)
%      Prad = Prad + U(i,j)*sin(T(j))*(pi/180)^2;
%   end
%end

DM=4*pi*U./Prad; %DM a field of Directivity Magnitudes normalized according to
                 % equation 2-16 on page 39 of Balanis.

D0 = max(max(DM)); % Directivity is the maximum value in this field.

%Elevation vs. Directivity
M1=DM(91,:); %pluck out the graph that sweeps theta while phi=90
figure; polar(T,M1);title('Elevation Plane
');


%Azimuth vs. Directivity
M2=DM(:,91); %pluck out the graph that sweeps  phi  while theta=90
figure; polar(P,M2');title('Azmuthal Plane
');

%Plot the whole Directivity Magnitude field.
figure; mesh(DM);
xlabel('Elevation');
ylabel('Azimuth');
zlabel('Intensity (U)');
title(ant_length/lamda);

Rrad = 2*Prad/max_current^2;

%Reflect the three desired parameters back to the user:
D0
Prad
Rrad
```

ee540_dipole_3.m

```
function ee540_dipole_3
%ee540_dipole3 --
%        Calculate the Directivity (D0), Radiation Resistance (Rrad), and
%        Radiated Power (prad) for a given wavelength (lambda), dipole length,
%        and dipole diameter.
%
%        This calculation is for a dipole antenna of any length,
%        centered at the origin and running along the z-axis.
%


% Prompt the User for lambda, antenna length (in terms of lambda),
%                     antenna radius (in terms of lambda), the number
%                     of current steps, and feed voltage.
lambda = input('Wave Length (default=500): ');
if isempty(lambda)
   lambda = 500;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%default value for lambda = 500
end

multiplier = input('\n\nAntenna Length as a fraction of Wavelength\n   (.5 for 1/2
wavelength)\n   (.005 for an infintesimal dipole)\n   (1.25 to see some side
lobes)\n\nEnter Length Fraction (default=.5): ');
if isempty(multiplier)
   multiplier = .5;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%default value for mutliplier = 0.5
end
ant_length = multiplier*lambda;  %calculate the length of the antenna

multiplier2 = input('\n\nAntenna Radius as a fraction of Wavelength\n   (.005 for thin
wire)\n   1/100 of dipole lenght is recommended \n\nEnter Radius Fraction (default=.005):
');
if isempty(multiplier2)
   multiplier2 = .005;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%default value for mutliplier2 = 0.005
end
wire_radius = multiplier2*lambda;  %calculate the diameter of the antenna

current_steps = input('\nNumber of constant current segments to use in approximating\n
the current (must be an odd number: default=199): ');
if isempty(current_steps)
   current_steps=199; %%%%%%%%%%%%%%%%%%%%%%%%%%%%%default value for current_steps = 199
end
if (mod(current_steps,2)==1)
   current_steps = current_steps + 1; % ensure an odd number of segments & even number of
points
end

feed_voltage = input('\nPeak Voltage at Feed (default=.01*j): ');
if isempty(feed_voltage)
   feed_voltage = .01*j;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%default value for feed_voltage
end


%k is the wave number (called beta in some contexts).
k=2*pi/lambda;


%zprime holds the z-locations for the end of each current segment.
zprime=linspace(-ant_length/2,ant_length/2,current_steps);

%current holds an approximation of the current along each segment of the antenna.
%this code uses equation
current=calc_current3(lambda, ant_length, wire_radius, zprime, feed_voltage);


%plot the current distribution along the antenna
figure; plot(abs(current),zprime);  title('Current Distribution (Magnitude)');
xlabel('Current (amps)'); ylabel('Z Location (m)');
```

```
%plot the current distribution along the antenna
figure; plot(angle(current),zprime);  title('Current Distribution (Phase)');
xlabel('Current (amps)'); ylabel('Z Location (m)');


prad = 0;

T=[0:2*pi/360:2*pi];
P=[0:2*pi/360:2*pi];

F = dipoleint2(T, P, lambda, current, zprime);

%return

%Calculate the total power radiated by "integrating" all values of the
% radiation intensity times sin(theta)*dTheta*dphi
% from theta = 0 to theta = pi;
% from phi=0 to 2*pi;
semicircle = sin(T(1:181)); %semicircle is a vector of sin(theta) from 0 to pi
prad1 = F(181,1:181); %at this point, prad1 sweeps through the ?? Field from theta = 0:pi
prad1 = prad1.*(pi/180).*semicircle; %now prad is F*sin(theta)*dtheta
prad = sum(prad1)*2*pi; %prad is the "area" under prad1 rotated 2*pi around phi.

%same prad calculation using loops for the double integration.
%prad=0;
%for (i=1:361)
%   for (j=1:181)
%      prad = prad + F(i,j)*sin(T(j))*(pi/180)^2;
%   end
%end

M=4*pi*F./prad;

M1=M(91,:); %pluck out the graph that sweeps theta while phi=90
M2=M(:,91); %pluck out the graph that sweeps  phi  while theta=90

figure; polar(T,M1);title('Elevation Plane
');

figure; polar(P,M2');title('Azmuthal Plane
');

D0 = max(max(M));

Rrad = max(current);
Rrad = abs(2*prad/Rrad^2);

D0
prad
Rrad

figure; mesh(M);
xlabel('Elevation');
ylabel('Azimuth');
zlabel('Intensity (U)');
title(ant_length/lambda);
%keyboard
```

ee540_array.m

```
function ee540_array
%ee540_array --
%        Calculate the Directivity (D0), Radiation Resistance (Rrad), and
%        Radiated Power (prad) for an array of dipole antennas each with a
%        given wavelength (lambda), dipole length, and dipole radius. The user
%        is asked to provide the number of elements in the array, the distance
%        between these elements, and the phase shift in current by which each
%        element leads its predecessor.
%
%        This calculation is for an array of dipole antennas of any length,
%        centered at the origin and running along the z-axis.
%


% Prompt the User for lambda, antenna length (in terms of lambda),
%                     antenna radius (in terms of lambda), the number
%                     of current steps, and feed voltage.
nelem = input('Number of dipoles in this linear array (default=7): ');
if isempty(nelem)
   nelem=7;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%default value for nelem = 7
end

beta = input('\nPhase by which each dipole`s current leads its predecessor (default=0
radians): ');
if isempty(beta)
   beta = 0;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%default value for beta = 0
end

lambda = input('\nWave Length (default=500): ');
if isempty(lambda)
   lambda = 500;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%default value for lambda = 500
end

dist = input('\nDistance between dipoles as a multiple of Wavelength (default=2): ');
if isempty(dist)
   dist = 2;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%default value for dist = 2
end

multiplier = input('\nAntenna Length as a fraction of Wavelength\n   (.5 for 1/2
wavelength)\n   (.005 for an infintesimal dipole)\n   (1.25 to see some side
lobes)\n\nEnter Length Fraction (default=.5): ');
if isempty(multiplier)
   multiplier = .5;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%default value for mutliplier = 0.5
end
ant_length = multiplier*lambda;  %calculate the length of the antenna

multiplier2 = input('\nAntenna Radius as a fraction of Wavelength\n   (.005 for thin
wire)\n   1/100 of dipole lenght is recommended \n\nEnter Radius Fraction (default=.005):
');
if isempty(multiplier2)
   multiplier2 = .005;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%default value for mutliplier2 = 0.005
end
wire_radius = multiplier2*lambda;  %calculate the diameter of the antenna

current_steps = input('\nNumber of constant current segments to use in approximating\n
the current (must be an odd number: default=199): ');
if isempty(current_steps)
   current_steps=199; %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%default value for current_steps = 199
end
if (mod(current_steps,2)==1)
   current_steps = current_steps + 1; % ensure an odd number of segments & even number of
points
end

feed_voltage = input('\nPeak Voltage at Feed (default=.01*j): ');
if isempty(feed_voltage)
   feed_voltage = .01*j;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%default value for feed_voltage
end
```

```
%k is the wave number (called beta in some contexts).
k=2*pi/lambda;

T=[0:2*pi/360:2*pi];
P=[0:2*pi/360:2*pi];

%get the Array Factor vs. Phi
AF=array_factor(P,lambda,nelem,beta,dist);


%plot the array factor's magnitude vs. phi.
figure; polar(P,AF);  title('Array Factor
');
%xlabel('Current (amps)'); ylabel('Z Location (m)');


%zprime holds the z-locations for the end of each current segment.
zprime=linspace(-ant_length/2,ant_length/2,current_steps);

%current holds an approximation of the current along each segment of the antenna.
%this code uses equation
current=calc_current3(lambda, ant_length, wire_radius, zprime, feed_voltage);


%plot the current distribution along the antenna
figure; plot(abs(current),zprime);  title('Current Distribution (Magnitude)');
xlabel('Current (amps)'); ylabel('Z Location (m)');

%plot the current distribution along the antenna
figure; plot(angle(current),zprime);  title('Current Distribution (Phase)');
xlabel('Current (amps)'); ylabel('Z Location (m)');


prad = 0;

%get the Radiation Intensity
U = dipoleint_array(T, P, lambda, current, zprime, AF);

%keyboard;

%Calculate the total power radiated by "integrating" all values of the
% radiation intensity times sin(theta)*dTheta*dphi
% from theta = 0 to theta = pi;
% from phi=0 to 2*pi;
semicircle = sin(T(1:181)); %semicircle is a vector of sin(theta) from 0 to pi
prad1 = U(181,1:181); %at this point, prad1 sweeps through the ?? Field from theta = 0:pi
prad1 = prad1.*(pi/180).*semicircle; %now prad is U*sin(theta)*dtheta
prad = sum(prad1)*2*pi; %prad is the "area" under prad1 rotated 2*pi around phi.

%same prad calculation using loops for the double integration.
%prad=0;
%for (i=1:361)
%   for (j=1:181)
%      prad = prad + U(i,j)*sin(T(j))*(pi/180)^2;
%   end
%end

M=4*pi*U./prad;

M1=M(91,:); %pluck out the graph that sweeps theta while phi=90
M2=M(:,91); %pluck out the graph that sweeps  phi  while theta=90

figure; polar(T,M1);title('Elevation Plane
');

figure; polar(P,M2');title('Azmuthal Plane
');

D0 = max(max(M));
```

```
Rrad = max(current)*nelem; %add the parrallel currents of all array elements
Rrad = abs(2*prad/Rrad^2);

D0
prad
Rrad

figure; mesh(M);
xlabel('Elevation');
ylabel('Azimuth');
zlabel('Intensity (U)');
title(ant_length/lambda);
%keyboard
```

dipoleint1.m

```
function [F]=dipoleint(theta, phi, lamda, length1, current)
% F = dipoleant(theta, phi, lamda, length1, current)
%calculate the radiation intensity of a dipole antenna using equations
% 4-62 on page 153 and 2-12a on page 38 of Balanis.
%   theta = elevation angle
%     phi = azimuthal angle
%  lambda = wave length (in meters)
% length1 = antenna length (in meters).
% current = maximum current (in amps).

     mu0 = 4*pi*10^-7;         % permeability (in newtons/amp^2)
epsilon0 = 8.854187817*10^-12; % permittivity (in Farads/meter )
     eta = sqrt(mu0/epsilon0); % intrinsic impedance (in ohms)

     beta = (2*pi)/lamda;       %wave number (in radians/meter)

     P1 = ones(size(phi));

    [MT MP] = meshgrid(theta, P1); % P1 is all ones for this case

%     keyboard

n1=cos((beta*length1/2).*cos(MT))-cos(beta*length1/2);
denom = (2*pi.*sin(MT+.0000000001));
% .0000000001; %avoid divide by zero by ensuring sin() <> 0
eTheta = j*eta*current.*n1./denom;
%ePhi = eTheta; %ePhi = eta*hPhi and hPhi = eTheta/eta ePhi=(eta/eta)*eTheta
%ePhi = 0 for a dipole antenna.
ePhi = 0;
F = (abs(eTheta).^2 + abs(ePhi).^2)/(2*eta);
```

dipoleint2.m

```
function [F]=dipoleint2(theta, phi, lamda, current, zprime)
% F = dipoleant(theta, phi, lamda, length1, current)
%calculate the radiation intensity of a dipole antenna using equations
% 4-58a on page 152 and 2-12a on page 38 of Balanis.
%   theta = elevation angle
%     phi = azimuthal angle
%  lambda = wave length (in meters)
% current = range of current (in amps).
% zprime  = range of z-locations allong the antenna each corresponding to an element of
current

     mu0 = 4*pi*10^-7;         % permeability (in newtons/amp^2)
epsilon0 = 8.854187817*10^-12; % permittivity (in Farads/meter )
     eta = sqrt(mu0/epsilon0); % intrinsic impedance (in ohms)
     beta = (2*pi)/lamda;       %wave number (in radians/meter)

     P1 = ones(size(phi));
     MT = theta*P1';       %Create a 361x361 array that varies in theta, but not phi.
     % could also use meshgrid to fill up MT
     %[MT MP] = meshgrid(theta, P1); % P1 is all ones for this case


%integrate E-theta over the whole length of the antenna for each theta
% using equation 4-58a on page 152 of Balanis.
n1=j*eta*beta/(4*pi); % constant in front of the integral
L1=size(theta');   L1 = ceil(L1(1)/2);
dz=zprime(2)-zprime(1);

for jj=1:L1 % for each value of theta from 0 to pi
%integrate the effects of the entire current distribution along the antenna.
int1=sum(current.*exp(j*beta*zprime*cos(theta(jj))))*dz;
 eTheta(1:361,jj) = n1*sin(theta(jj))*int1;
 eTheta(1:361,362-jj) = n1*sin(theta(jj))*int1; % fill in the full circle so that
 % we can make nice looking elevation-plane plots of the field.
end

ePhi = 0;
% calculate the radiation intensity using equation 2-12a on page 38
F = (abs(eTheta).^2 + abs(ePhi).^2)/(2*eta);
```

dipoleint_array.m

```
function [F]=dipoleint_array(theta, phi, lamda, current, zprime, AF)
% F = dipoleant(theta, phi, lamda, length1, current)
%calculate the radiation intensity of a dipole antenna using equations
% 4-58a on page 152 and 2-12a on page 38 of Balanis.
%   theta = elevation angle
%     phi = azimuthal angle
%  lambda = wave length (in meters)
% current = range of current (in amps).
%  zprime = range of z-locations allong the antenna each corresponding to an element of
current
%      AF = Array Factor to apply to the E-field.

     mu0 = 4*pi*10^-7;         % permeability (in newtons/amp^2)
epsilon0 = 8.854187817*10^-12; % permittivity (in Farads/meter )
     eta = sqrt(mu0/epsilon0); % intrinsic impedance (in ohms)
     beta = (2*pi)/lamda;       %wave number (in radians/meter)

     P1 = ones(size(phi));
     MT = theta*P1';       %Create a 361x361 array that varies in theta, but not phi.
     % could also use meshgrid to fill up MT
     %[MT MP] = meshgrid(theta, P1); % P1 is all ones for this case


%integrate E-theta over the whole length of the antenna for each theta
% using equation 4-58a on page 152 of Balanis.
n1=j*eta*beta/(4*pi); % constant in front of the integral
L1=size(theta');   L1 = ceil(L1(1)/2);
dz=zprime(2)-zprime(1);

for jj=1:L1 % for each value of theta from 0 to pi

%integrate the effects of the entire current distribution along the antenna.
int1=sum(current.*exp(j*beta*zprime*cos(theta(jj))))*dz;

 eTheta(1:361,jj) = (n1*sin(theta(jj))*int1);
 eTheta(1:361,362-jj) = (n1*sin(theta(jj))*int1); % fill in the full circle so that
 % we can make nice looking elevation-plane plots of the field.
end

t=ones(size(AF));
tt = t'*AF; % make AF fill 3-space so we can multiply eTheta;


eTheta=tt.*eTheta;

ePhi = 0;

% calculate the radiation intensity using equation 2-12a on page 38
F = (abs(eTheta).^2 + abs(ePhi).^2)/(2*eta);
```

pocklington.m

```
function [P]=pocklington(r, radius, lambda)
%      P = pocklington(r, radius)
%      r = distance from this point to the skin of the wire of the segment
%          being evalutated. r may be a vector of such distances.
% radius = radius of the wire being analyzed
% lambda = wavelength
%
% This function returns the integrand for each segment of a
% wire being evalutated using Pocklington's Ingetral Equation. If the
% input parameter r contains a vector of distances, then P is returned as
% a vector of integrand values. If r is a scalar value then, P is returned
% as a scalar.

     mu0 = 4*pi*10^-7;         % permeability (in newtons/amp^2)
epsilon0 = 8.854187817*10^-12; % permittivity (in Farads/meter )
     eta = sqrt(mu0/epsilon0); % intrinsic impedance (in ohms)
     beta = (2*pi)/lambda;      % wave number (in radians/meter)

   P = exp(-j*beta*r).*((1+j*beta*r).*(2*r.^2-3*radius^2)+(beta*radius*r).^2);
   P = P./(4*pi*r.^5);
```

calc_current.m

```
function [current]=calc_current(lambda, length2,wire_radius,zprime,feed_voltage)
%function [current]=calc_current(length,radius,zprime,feed_voltage)
% lambda = wavelength
% length = length (in terms of wavelength) of the dipole
% radius = radius (in terms of wavelength) of the dipole
% zprime = array of points along the z axis for which the current is to be calculated.
% feed_voltage = peak voltage at antenna feed.
%
% Solve for I in Z*I=V
%   Where I is the unknown current distribution
%         Z is a square matrix filled using Pocklington's Integral Equation
%         V is the user defined voltage (Becuase the antenna is center fed,
%             only the middle entry has a value all other entries are zero.)


     mu0 = 4*pi*10^-7;         % permeability (in newtons/amp^2)
epsilon0 = 8.854187817*10^-12; % permittivity (in Farads/meter )
     eta = sqrt(mu0/epsilon0); % intrinsic impedance (in ohms)
     beta = (2*pi)/lambda;      % wave number (in radians/meter)

current_steps = length(zprime); % of points for which to find a current
%
% 1     2     3     4     5     6     7     8     9     A     B     C  Point
% *-----*-----*-----*-----*-----*=====*-----*-----*-----*-----*-----*
%    1     2     3     4     5     6     7     8     9     A     B     Segment
%                               ^
%                             gap
% The points of zprime are at the ends of each segment.
% zprime=linspace(-length2/2,length2/2,current_steps);
% The current is calculated at the center of each segment.
% This creates an index mismatch.

ie_steps = 199;
dz = zprime(2) - zprime(1);
ddz = dz/ie_steps;

% Caclulate a new vector with points at the center of each segment
%zcenter = zprime+.5*dz;
%zcenter = zcenter(1:length(zcenter)-1); %chop off the last element
%zdist = zcenter-zcenter(1); % distance from the end segment to every other segment
%rr = sqrt(wire_radius^2 + zdist.^2);

ZM = length2/2-.5*dz;
      jj=1:ie_steps;
      z = -.5*dz + ddz*jj;
for ii=1:current_steps-1
  % crt = 0;
   ZN(ii) = length2/2 -(ii-.5)*dz;
      r = sqrt(wire_radius^2 + (ZN(ii)-ZM+z).^2); %distance from segment ii to skin of jj
      crt = pocklington(r,wire_radius, lambda)*ddz;
   p(ii) = sum(crt);
end

%
% Mark This code works without loops, but introduces a numerical "wobble" that throws-off
% the final directivity calculation by .02. I am sticking with the loop.--Rob
%keyboard
%P=p;
%ii=1:(current_steps-1);
%ZM = ZM+z;
%ZN=length2/2-(ii-.5)*dz;
%ZZ = ones(size(ii));
%ZZN = ZZ'*ZN;
%ZZM = ZM'*ZZ;
%r = sqrt(wire_radius^2 + (ZZN-ZZM).^2);
%crt = pocklington(r,wire_radius, lambda)*ddz;
%p = sum(crt);
%figure; plot(p-P); title('wobble P-p');
```

```
% Because Z is a toeplitz matrix we can fill in the entire matrix now that we have the
%  first row. The general pattern is:
%
%             |A B C D E F G . . . Z|
%             |B A B C D E F . . . Y|
%             |c B A B C D E . . . X|
%             |.             .     .|
%             |.               .   .|
%             |.                 . .|
%             |Z Y X W V U T . . . A|
znm=toeplitz(p');


middle = floor(current_steps/2);
G = zeros(size(p));
G(middle) = feed_voltage;

current=G/znm;
%
%
% 1     2     3     4     5     6     7     8     9     A     B     C  Point
% *-----*-----*-----*-----*-----*=====*-----*-----*-----*-----*-----*
%    1     2     3     4     5     6     7     8     9     A     B     Segment
%                                  ^
%                                gap
% The points of zprime are at the ends of each segment.
% zprime=linspace(-length2/2,length2/2,current_steps);
% The current is calculated at the center of each segment.
% This creates an index mismatch.
%
%
% A previous version of this algorithm shifted the 2nd 1/2 of the segment values
% to the right and added a new current value in the center (point 7 in the sketch).
% This lead to numerical instability in the phase of the current producing asymetry
% about the XY plane--very bad!
%                                gap
% 1     2     3     4     5     6  V  7     8     9     A     B     C  Point
% *-----*-----*-----*-----*-----*=====*-----*-----*-----*-----*-----*
%  ^     ^     ^     ^     ^     ^     ^     ^     ^     ^     ^     ^  UNSTABLE PHASE!!!
%   \     \     \     \     \     \  /     /     /     /     /     /
%    1     2     3     4     5     6     7     8     9     A     B     Segment
%
%L = length(current)+1;
%current(L)=1; % add the last point back to the array
%temp = current(middle:L-1);
%current(middle+1:L)=temp;
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% This version mirrors the uper 1/2 of the dipole into the lower 1/2. Thus ensuring
% symetry--very good. The next five lines of code increase the size of the CURRENT
% vector by one (to account for the "lost" point) then mirrors the first 1/2 of the
% vector into the 2nd 1/2.
%This is done using the MatLab FLIPDIM command and some creative indexing.
%                                gap
% 1     2     3     4     5     6  V  7     8     9     A     B     C  Point
% *-----*-----*-----*-----*-----*=====*-----*-----*-----*-----*-----*
%  ^     ^     ^     ^     ^     ^     ^     ^     ^     ^     ^     ^
%   \     \     \     \     \     \  /      |     |     |     |     |
%    1     2     3     4     5     6        |     |     |     |    /   Segment
%     \     \     \     \     \_____________/     /     /     /    /
%      \     \     \     \_____________________/     /     /    /
%       \     \     \_____________________________/     /    /
%        \     \_____________________________________/    /
%         \_____________________________________________/
%
%
L = length(current)+1;
current(L)=1; % add the last point back to the array
temp = current(1:middle);
current(middle+1:L)=flipdim(temp,2);
```

array_factor.m

```
function [AF]=array_factor(theta, lambda, nelem, beta, dist)
% AF = array_factor(theta, lamda, nelem)
%calculate the broadside array factor of a linear array of antennas using equation
% 6-10a on page 259 of Balanis.
%   theta = azimuthal angle measured in the plane normal to the dipole antennas
%           and containing the centers of the antennas.
%  lambda = wave length (in meters)
%   nelem = number of antennas in the array
%    beta = phase shift by which each element's current leads the current of its
predecessor
%    dist = distance betwen successive elements of the array (as multiple of lambda).

     mu0 = 4*pi*10^-7;         % permeability (in newtons/amp^2)
epsilon0 = 8.854187817*10^-12; % permittivity (in Farads/meter )
     eta = sqrt(mu0/epsilon0); % intrinsic impedance (in ohms)
     k = (2*pi)/lambda;        % wave number (in radians/meter)
     d = dist*lambda;          % distance between antennas

if((d~=0)|(nelem~=1)) %check for single element array
   psi = k*d*cos(theta)+beta;

     AF=sin(nelem*psi/2)./(nelem*sin(psi/2));
  else
     AF=ones(size(theta));
  end
```