

Welcome

Welcome to Arrayfeed1. This program is copyright © 2007 by Roy W. Lewallen. It may be freely copied and distributed, providing no fee is charged. The user assumes all responsibility for its use. No warrantee of any kind is given or implied.

Introduction

What is Arrayfeed1?

Correctly feeding a phased array is usually more difficult than just cutting “phasing lines” for the desired phase delay between elements. The reasons are explained briefly in How Does Arrayfeed1 Work? and in more detail in the references in that topic.

Arrayfeed1 makes calculations necessary to design feed systems for phased arrays which actually produce the desired pattern. It makes calculations for two kinds of phased array feed systems for three types of popular arrays. It can be extended to use for other arrays consisting of groups of in-phase or 180 degree out of phase elements to be phased relative to each other.

The first feed system type is the "simplest" system which consists of only feedlines, resembling a “phasing line” feed system. The difference is that, unlike a simple “phasing line” system, the line lengths calculated by Arrayfeed1 actually produce the desired current magnitude ratio and phase difference in the elements. The second type is an L network feed system consisting of two quarter wavelength lines and an L network. This has the advantage of being more readily adjustable than the “simplest” system.

The program directly calculates the required line lengths or network component values for a 2 element, 4 square, or four element rectangular array. Although diagrams and most text refer to base-fed vertical elements, the program will work with any kind of elements, including such types as loops. The elements should generally be identical. However, it can be used for arrays with some different sized elements provided that certain elements are identical. See Array Types for more information.

Arrayfeed1 was originally developed to accompany the 21st Edition of the ARRL Antenna Book. You can find much more information about its use, and several detailed examples, in Chapter 8 of the Antenna Book, 21st Edition and later.

Array Types

Arrayfeed1 can calculate “simplest” line lengths and L network component values for three basic array types. Elements can be any type (e.g., base fed vertical, dipole, loop, etc.) and don't have to be identical. Using the “simplest” system line lengths or L network component values calculated by Arrayfeed1 will produce the specified element current ratio. However, it won't guarantee a particular pattern. What you have to do is to first get the desired pattern with a modeling program and from that determine what currents are needed. Arrayfeed1 will then calculate the necessary line lengths or component values.

The three basic array types are:

1. Two element array. The feedline lengths can be calculated for any array of two elements, for any current ratio (magnitude and phase) and element spacing.
2. Four square array. The classic four square array has four elements in a square pattern, a quarter wavelength on a side. The maximum pattern direction is along a diagonal. Relative to the element toward the rear of the pattern, the element toward the front has a current phase angle of -180 degrees, the two side elements have a current phase angle of -90 degrees, and the magnitudes of all element currents are the same. Arrayfeed1 can calculate line lengths for this array, or certain variations. The main requirements are that the front and rear elements must have equal magnitude currents which are 180 degrees out of phase, and the two side elements be identical and have currents which are the same magnitude and phase as each other. Elements can have any desired spacing (provided symmetry is retained – see below), and you can specify any relationship between the current magnitude and phase in the side elements relative to the front and rear elements. This allows you to design the wider spaced arrays which are sometimes used, or experiment with different spacing/phasing combinations.
3. Four element rectangular array. The most common manifestation of this array has two elements spaced a half wavelength apart and fed with equal, in phase currents; and another two elements also fed in phase with each other but spaced a quarter wavelength from the first pair and fed with currents 90 degrees delayed from the first pair. Arrayfeed1 requires only that each pair of elements be identical and have equal, in phase currents. You can specify any magnitude and phase for the current in the second pair relative to the first. Any spacing can be used, provided that symmetry is retained as described below.

As the program is currently written, the four element arrays have to retain symmetry, because it assumes the side elements of the four square array, or each pair of elements in the rectangular array, have the same feedpoint impedance. This means that you can, for example, move the elements in the four square to make a sharper diamond shape, or the elements in the rectangular array to make a square or different ratioed rectangle. But you can't analyze a parallelogram shaped array with the rectangular array option because the feedpoint impedances of the two elements of each pair would be different.

How Does Arrayfeed1 Work?

Proper operation of phased arrays requires that the currents at the element feedpoints have a particular relationship. That is, each element current must have a particular magnitude and phase angle compared to some reference. This is often done by using a “phasing line” from some common point to each element, with the line providing the required phase shift for each element.

But although this method often works well for very large arrays, it generally doesn't work for arrays of a few elements. The reason is that the delay, of voltage or current, in a transmission line equals the electrical length of the line only if the line is terminated in its characteristic impedance. In all other cases except for a very few special cases, the delay will not be equal to the electrical length of the line. In fact, it can often differ by a large amount. In addition, a line not terminated in its characteristic impedance will transform the voltage and current, making them greater or less at one end of the line than the other. (Of course, if the voltage is greater, the current is less or vice-versa.) So when a “phasing line” is connected to an element whose feedpoint impedance doesn't equal its characteristic

impedance, it produces not only the wrong phase shift, but also the wrong current magnitude. This subject is discussed in detail in Chapter 8 of the ARRL Antenna Book.

Two methods which can be used to produce the desired currents in the array elements are the "simplest" system and an L network feed system. Arrayfeed1 uses basic transmission line equations to calculate either the line lengths necessary for the "simplest" system or component values for an L network feed system.

The design of any phased array feed system providing currents to the elements which aren't exactly in phase or 180 degrees out of phase requires knowledge of some or all the element feedpoint impedances when the array is being correctly fed. These can most easily be determined by modeling the correctly fed antenna with EZNEC or a similar modeling program. For more information about EZNEC or to obtain the demo program see <http://eznec.com>.

Using Arrayfeed1

Specifying Inputs

Arrayfeed1 requires that some or all the feedpoint impedances be known. These are obtained with a modeling program such as EZNEC. The procedure is to model the antenna, with a current source at each element feedpoint. The currents are adjusted to obtain the desired pattern. Then the feedpoint impedances and currents are read from the modeling program and entered into Arrayfeed1.

Inputs required by Arrayfeed1 are:

- Element feedpoint impedances. Note that these are not the impedances of isolated elements, but the impedances of the array elements when all elements are being fed with normal array currents. You get these from a modeling program like EZNEC. In the four element arrays, some elements have the same feedpoint Z, so there's provision for only one of the identical elements. See the Example. Design of the "simplest" feed system requires knowledge of all the element impedances; L network design requires only the impedance of the lagging element(s). However, Arrayfeed1 will, if requested, also tell you the input impedance of the whole feed system. If you want this information, it's necessary to enter all the feedpoint impedances.
- Line impedances. For the "simplest" feed system, Arrayfeed1 will calculate the lengths of two "phasing lines", labeled "Line 1" and "Line 2" in the picture shown in the Arrayfeed1 window. You must choose the impedances of these lines. Sometimes, there is no solution to the problem with the specified current and line impedances. If that occurs, you can often get a solution by specifying different line impedances. The two line impedances don't have to be the same, but you should make them the same if you anticipate remotely switching the array direction. You're free to choose the impedances for the two element L network feed system $\frac{1}{4}$ or $\frac{3}{4}$ wavelength lines, but again make them the same if you'll be doing remote directional switching. In L network systems for larger arrays, the current "forcing" lines for the element groups are used in place of the $\frac{1}{4}$ or $\frac{3}{4}$ wavelength lines, so this input doesn't appear separately. For either feed system, line impedances drastically different from the element feedpoint impedances will generally result in arrays more sensitive to small frequency and physical changes.

- Z0 of $\frac{1}{4}$ or $\frac{3}{4}$ wavelength lines. This entry doesn't appear when the two element array type is chosen. The four element arrays each have four additional transmission lines, marked on the picture with an asterisk (*). These four lines must each be $\frac{1}{4}$ or $\frac{3}{4}$ electrical wavelengths long (or any odd multiple of $\frac{1}{4}$ wavelength), and all must be the same length and have the same Z0. You must choose the Z0 of these lines. If Arrayfeed1 can't find a solution for a "simplest" feed system, you might be able to get a solution by changing the specified Z0 of these lines. Usually, $\frac{1}{4}$ wavelength lines are better than $\frac{3}{4}$ because then the array characteristics will change less with frequency. However, in some cases $\frac{1}{4}$ wavelength lines aren't physically long enough. For example, $\frac{1}{4}$ wavelength of solid polyethylene insulated coax, which has a velocity factor of 0.66, is $0.66 * 0.25 = 0.165$ wavelength. Two of these extend only 0.33 physical wavelengths, which isn't far enough to reach between two elements of a rectangular array if the elements are spaced $\frac{1}{2}$ wavelength apart. So $\frac{3}{4}$ wavelength lines would have to be used in that case. There also might be some cases where longer lines will result in an array pattern or input impedance which changes less or in a more favorable direction with frequency. Experimentation with different lengths will be necessary to find the best length for a given array and application. Remember, though, that all four of these lines must be the same length. The four square feed system has one additional line which is a half wavelength long and can be any Z0. There is no input specification for this line.
- Current ratio. For the two element array, these values are the ratio of current in the lagging element relative to the leading element. For example, if the leading element is to have a current of 1 amp and a phase angle of 0 degrees, and the lagging element is to have a relative current of 1.2 amps and a phase angle of -55 degrees, you would enter 1.2 for the magnitude and -55 for the phase. For the four square array, these values are the ratio of current in the two side elements to the leading element. (The lagging element is always equal in magnitude and 180 degrees out of phase relative to the leading element.) Specify a magnitude of 1 and phase of -90 degrees for a classical four square array. For the rectangular array, these values are the current in the two lagging elements relative to the two leading elements.

Optional inputs are frequency in MHz and line velocity factor. Frequency and velocity factor are needed if you want to know the physical lengths of "simplest" feed system lines or $\frac{1}{4}$, $\frac{1}{2}$, or $\frac{3}{4}$ wavelength lines, and frequency is necessary to compute L and C values of L network components. If your array uses lines having different velocity factors, enter the velocity factor for one of the line types and read its physical length. Then change the velocity factor to another of the line types and read its physical length, and so forth.

Detailed examples showing design of both feed systems with each of the array types can be found in the ARRL Antenna Book, 21st Edition and later.

Getting a Solution

When you click the Find Solutions button, one of three things will happen.

1. Values appear in the Solutions section. The values in the Solutions section depend on the specified feed system type. For the "simplest" system, the Solutions results are the lengths, in electrical degrees, of the two lines Line 1 and Line 2 necessary to feed the array. For the L network feed system, the values and types of L network components are shown. Additional information appears in the Physical Lengths section if you've entered a velocity factor for the

lines. For the “simplest” system, the physical lengths of the “phasing” lines are shown. For both feed system types, lengths of $\frac{1}{4}$, $\frac{1}{2}$, and $\frac{3}{4}$ wavelength lines are given.

2. No Solution appears in the Solutions section. This can occur only with a “simplest” feed system design, because there is no “simplest” system solution to the problem for some combinations of transmission line impedances for a given set of element feedpoint impedances. If No Solution appears, try changing the Line 1 and Line 2 Z0 specifications, and for the four element arrays, the Z0 of the $\frac{1}{4}$ or $\frac{3}{4}$ wavelength lines. They don’t have to be equal. Usually you can find a combination of common impedances (such as 50 and 75 ohms) which will produce a solution. If you can’t, this feed method can’t be used for this array.
3. You get an message telling you that the value in one of the input boxes is invalid. Correct the problem and continue. Check the other boxes also; only the first error found is reported, so if there are multiple errors, you’ll get a message each time you attempt a solution until all errors are fixed.

The “simplest” feed system always produces two solutions. Although it’s tempting to select the solution providing the best impedance match to the main feedline, this isn’t a good criterion to use. Usually, it’s best to choose the solution in which the difference in electrical lengths between the lines is closest to the phase delay of the currents between the elements. This will usually result in an array whose pattern and impedance are less sensitive to frequency and physical array variations. Impedance matching can be done at the main feedpoint as necessary.

Example

This example of program operation for designing a classic four square array feed system uses EZNEC or the free EZNEC demo program available from <http://eznec.com> to find the feedpoint impedances and test the results. Both a “simplest” feed system and L network system are designed in this example.

Note that the values you see with either program might differ slightly from those shown here, due to slightly different calculation orders or methods in various program versions.

1. Start EZNEC or the EZNEC demo. Click Open on the left side of the main window. Select 4Square (EZNEC) or d_4Square (EZNEC demo) then click Open in the dialog box. This is a model of a classic four square array. (If using the demo, ignore the segmentation warnings.) The demo version will report slightly different values than the standard program because it’s limited to fewer than the optimum number of segments for this antenna.
2. Click View Ant at the left side of the main window to show a diagram of the antenna. The four small circles at the bases of the elements are the current sources.
3. Click Sources in the white portion of the main window to see the magnitude and phase of each source. If you want to experiment with different patterns, this is where you would modify the sources to change the phasing or current magnitudes. Note that source 1 is in element 1, and it has the most positive phase angle (zero degrees). So element 1 is the leading element. Sources 2 and 3 are in elements 2 and 3 respectively, with phase angles of -90 degrees. Element 4 is the lagging element, with a phase angle of 180 (the same as -180) degrees. All currents are equal in

magnitude at 1 ampere. We'll use these values in the example, but in the future feel free to experiment.

4. Click FF Plot at the left of the main window to see the azimuth pattern at an elevation angle of 23 degrees, where this model's field strength is the greatest. Note the gain and front/back ratio of 5.33 dBi and 24.46 dB respectively (EZNEC) or 5.29 dBi and 23.83 dB. (EZNEC demo). The feed system designed with Arrayfeed1 should produce close to the same values, because it should produce very nearly the same element currents.
5. Click Src Dat at the left of the main window. In the window that opens, record the impedance shown for Source 1 (which is in wire 1, the leading element), Source 2 (which is in wire 2, one of the side elements), and Source 4 (which is in wire 4, the lagging element). You should have something like this:

(EZNEC)

Leading Element: $-1.388 - j20.29$

Side Elements: $40.92 - j21.92$

Lagging Element: $58.73 + j53.33$

(EZNEC demo)

Leading Element: $-1.223 - j21.92$

Side Elements: $42.54 - j22.48$

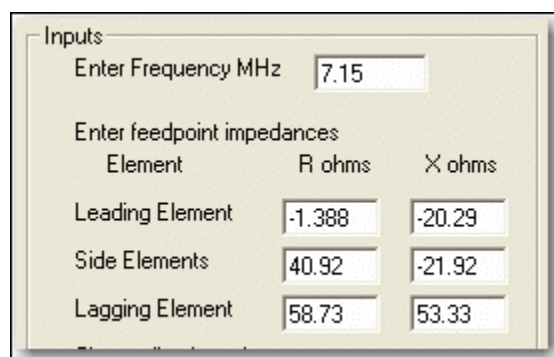
Lagging Element: $59.27 + j54.81$

This provides a good illustration of just how much the feedpoint impedances can change due to mutual coupling. The feedpoint impedance of an isolated single element is about $36 - j2$ ohms.

6. Now that you have the feedpoint impedances, you can end EZNEC. First the "simplest" feed system design will be illustrated, followed by the L network system design.

"Simplest" Feed System

1. Start Arrayfeed1.
2. In Arrayfeed1, select Four Square in the Array Type section and "Simplest" in the Feed System Type in the upper left. In the Inputs section, enter the frequency of 7.15 MHz, and the feedpoint impedances in the rows of boxes just below the frequency. If using EZNEC you should have:



Inputs		
Enter Frequency MHz	7.15	
Enter feedpoint impedances		
Element	R ohms	X ohms
Leading Element	-1.388	-20.29
Side Elements	40.92	-21.92
Lagging Element	58.73	53.33

3. We'll first try for a solution with all 50 ohm transmission lines. Enter 50 into the Line1 Z0 and Line 2 Z0 boxes, and in the $\frac{1}{4}$ or $\frac{3}{4}$ wavelength line Z0 box.
4. Enter 1 in the current Mag box and -90 in the current Phase box, for a classic 4 square.
5. Check the Calc Zin box if it isn't already checked, since we'll want to look at the overall array impedance. Then click the Find Solutions button. The result should be a red "No Solution" message in the Solutions section. This means that no solution can be found for the combination of feedpoint and transmission line impedances we've specified. Let's try changing the Line 1 and Line 2 Z0.
6. Change the Line 1 and Line 2 Z0 from 50 to 75 ohms and click Find Solutions again. You should now see solutions in the boxes in the Solutions section. Note that there are two sets of solutions – if any solution is found, there will always be two. You should have:

(EZNEC)

Line 1 = 24.38 deg, Line 2 = 168.21 deg

Line 2 = 115.56 deg, Line 2 = 212.68 deg

(EZNEC demo)

Line 1 = 24.05 deg, Line 2 = 168.45 deg

Line 2 = 116.65 deg, Line 2 = 211.41 deg

7. Now it's time to test the results. Start EZNEC or the EZNEC demo and open example file 4SqTL (EZNEC) or d_4SqTL (EZNEC demo). Click View Ant to see a diagram of the antenna. This is a model of a four square array using the feed system shown in the picture in the Arrayfeed1 window.
8. Click Trans Lines in the main EZNEC window to show the transmission lines. Transmission lines 1, 3, and 4 are three of the quarter wavelength lines marked with an asterisk (*) in the Arrayfeed1 picture. Transmission line 2 is the combination of the fourth quarter wavelength line and the half wavelength line. Transmission line 5 is Arrayfeed1's Line 1, and Transmission line 6 is Arrayfeed1's Line 2. Notice that the impedances of the lines are the same as the ones used for the Arrayfeed1 calculation.
9. We'll try the first Arrayfeed1 solution. In the EZNEC or EZNEC demo Transmission Lines window, in the No. 5 Length cell (current value 31.37 deg), enter 24.38d (EZNEC) or 24.05d (EZNEC demo) and press <Enter>. Don't forget the 'd', which tells EZNEC that the value is electrical degrees! In the No. 6 Length cell (current value 173.66 deg.), enter 168.21d (EZNEC) or 168.45d (EZNEC demo) and press <Enter>. Don't forget the 'd'!
10. In the EZNEC main window, click FF Plot. Note the gain and front/back ratio of 5.33 dBi and 24.48 dB (EZNEC) or 5.29 dBi and 23.83 dB (EZNEC demo). You can see that these are just about exactly the values we got for the array fed with separate current sources which we saw in step 4. This shows that the feed system is working as predicted. The element currents can be seen by clicking Currents in the main EZNEC window. From them, you can verify that side element 2 and 3 currents are lagging leading element 1 by very nearly 90 degrees, element 4 is lagging element 1 by very nearly 180 degrees, and that the magnitudes of all feedpoint currents

are very nearly equal. Try the Solution 2 line lengths if you'd like – you'll find equally good results.

11. In EZNEC, click Src Dat. This is the impedance seen at the array feedpoint. EZNEC reports $13.51 + j0.989$ ohms, Arrayfeed1 predicted $13.51 + j0.99$ for EZNEC values. EZNEC demo reports $13.31 + j0.8104$ ohms while Arrayfeed1 predicted $13.31 + j0.81$. This close agreement provides an additional check that values were entered correctly into both programs. It looks like a good match can be made to 50 ohm transmission line with only a 4:1 transformer. To check, change the Alt SWR Z0 in EZNEC or EZNEC demo's main window to 12.5 ohms to simulate the effect of a 4:1 transformer and 50 ohm line. In the Source Data window, note that the SWR for the 12.5 ohm system is about 1.1:1, a very good match. Of course, other matching methods could also be used.
12. Finally, let's make a model we can actually build. Going back to Arrayfeed1 (if you closed it, start it again and click Find Solutions). Enter .66 in the Velocity Factor box for solid polyethylene dielectric coax such as RG-8, RG-214, RG-58, etc. In Units, select Feet to make it agree with the EZNEC model used for the example. (You could, of course, change the units of the EZNEC model, but don't do that now, since the remainder of the example dimensions in feet.)
13. Looking at the Arrayfeed1 array diagram, note that the side elements are connected together by two lines having a total length of $\frac{1}{2}$ wavelength (if the lines are $\frac{1}{4}$ wavelength each) or 1.5 wavelengths (if the lines are $\frac{3}{4}$ wavelength each). In EZNEC, click Wires to open the Wires Window. From the coordinates of the elements, you can see that adjacent elements are spaced 34.42 feet apart, so the diagonal side elements are $34.42 * \sqrt{2} = 48.43$ feet apart. The two $\frac{1}{4}$ or $\frac{3}{4}$ wavelength lines have to reach between those two elements. In the Arrayfeed1 Physical Lengths section, look at the value in the 1/2 wavelength box, 45.396. This is the length in feet of $\frac{1}{2}$ electrical wavelength of transmission line having a velocity factor of 0.66 at a frequency of 7.15 MHz. So $\frac{1}{2}$ wavelength of solid dielectric coax would just reach between the two side elements. The front and rear elements have an extra half wavelength of line between them, so there won't be any problem with those reaching. We can use $\frac{1}{4}$ wavelength for the asterisk(*) – marked lines.
14. Now we'll put the physical lengths into the model and see what happens. In the Length column of the EZNEC Transmission Lines window, enter the following values for the four $\frac{1}{4}$ wavelength lines (remembering that transmission line 2 is a combination of $\frac{1}{4}$ and $\frac{1}{2}$ wavelength). Don't add 'd' this time, since the dimensions are feet, not degrees.

No.1 22.698
No. 2 68.093
No. 3 22.698
No. 4 22.698

15. Enter the physical lengths of the “phasing lines” line 1 and line 2, from the Arrayfeed1 Physical Lengths section, into the EZNEC Transmission Lines window as the lengths of transmission lines 5 and 6:

(EZNEC)
No. 5 6.15

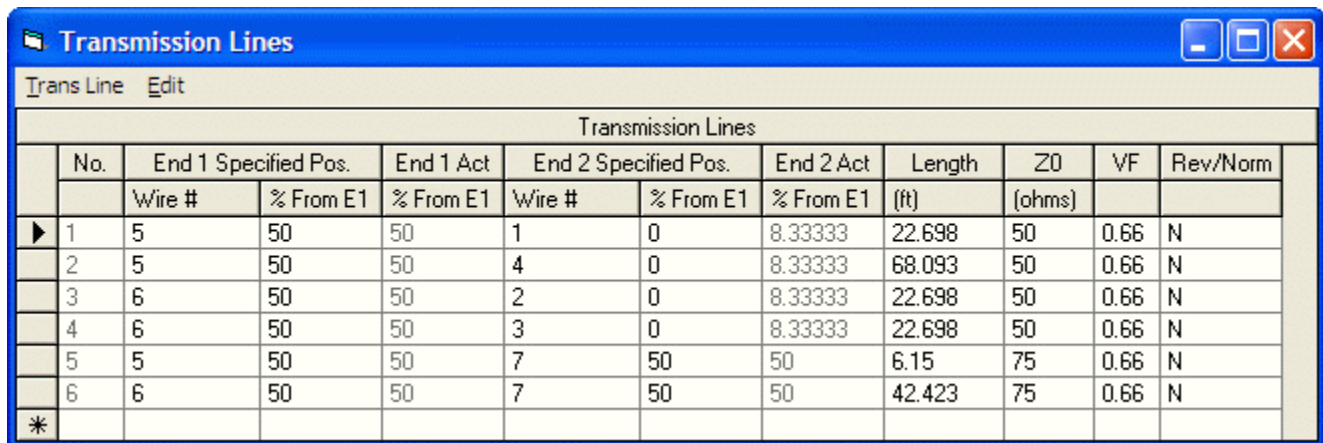
No. 6 42.423

(EZNEC demo)

No. 5 6.065

No. 6 42.482

16. In the EZNEC Transmission Lines Window, open the Trans Line menu at the top. Select Group Modify and click Ok. In any of the cells in the VF column change the value from 1 to .66 and press <Enter>. All VF values should now be 0.66. Open the Trans Line menu and click Group Modify again to turn this feature off. The EZNEC Transmission Lines Windows should now look like this (if using the demo program, the No. 5 and No. 6 Length entries should be 6.065 and 42.482 instead of the values shown):



The screenshot shows the 'Transmission Lines' window in EZNEC. The window title is 'Transmission Lines' and it has a menu bar with 'Trans Line' and 'Edit'. The main area contains a table with the following columns: No., End 1 Specified Pos., End 1 Act, End 2 Specified Pos., End 2 Act, Length, Z0, VF, and Rev/Norm. The table has 6 rows of data and a final row with an asterisk. The data is as follows:

No.	End 1 Specified Pos.		End 1 Act	End 2 Specified Pos.		End 2 Act	Length (ft)	Z0 (ohms)	VF	Rev/Norm
	Wire #	% From E1	% From E1	Wire #	% From E1	% From E1				
1	5	50	50	1	0	8.33333	22.698	50	0.66	N
2	5	50	50	4	0	8.33333	68.093	50	0.66	N
3	6	50	50	2	0	8.33333	22.698	50	0.66	N
4	6	50	50	3	0	8.33333	22.698	50	0.66	N
5	5	50	50	7	50	50	6.15	75	0.66	N
6	6	50	50	7	50	50	42.423	75	0.66	N
*										

17. Click FF Plot to see the pattern. You should see exactly or nearly exactly the same gain and front/back ratio you saw in steps 4 and 15. This model will accurately show the array pattern and feedpoint impedances at any frequency. You should check operation at the frequency extremes over which you expect to operate, and compare results using the two solutions. Note that the EZNEC 4 square model used for this example doesn't include any ground loss resistance. In a real design, the model should initially include approximate ground loss as a resistive load at the base of each element. (You'll find approximate values to use in Chapter 8 of the ARRL Antenna Book.) After the design is complete and you reach this point, you should change the model's loss resistance to the extremes which you feel the actual value could be, and recheck array performance with the designed feed system. One solution is likely to be more tolerant of variation than the other. This completes the "simplest" feed system design example.

L Network Feed System

1. Start Arrayfeed1 if not already started.
2. In Arrayfeed1, select Four Square in the Array Type section and L Network in the Feed System Type in the upper left. In the Inputs section, enter the frequency and feedpoint impedances as done in step 2 of the "simplest" design above. If you previously did the "simplest" example, the correct impedance values should already be present.

3. In this example, we'll use 50 ohms for the $\frac{1}{4}$ and $\frac{3}{4}$ wavelength lines, so enter 50 into the $\frac{1}{4}$ or $\frac{3}{4}$ wavelength line Z0 box if the value isn't already there.
4. Enter 1 in the current Mag box and -90 in the current Phase box if the values aren't already there.
5. Check the Calc Zin box if it isn't already checked, since we'll want to look at the overall array impedance. Then click the Find Solutions button. You should see the following results in the Solutions frame:

(EZNEC)

Xser = 0.68 uH (L)

Xsh = 1119.0 pF (C)

(EZNEC demo)

Xser = 0.654 uH (L)

Xsh = 1157.8 pF (C)

6. All that's necessary is to build the network according to the picture in Arrayfeed1. The Physical Lengths frame gives the lengths of the $\frac{1}{2}$ wavelength and $\frac{1}{4}$ or $\frac{3}{4}$ wavelength lines, once you enter the line velocity factor in the appropriate box. It's not possible to model the L network feed system with EZNEC through v. 4.0, but the system has been modeled by other means, and arrays have been built, measured, and used which incorporate this type of feed system. Note: EZNEC v. 5.0, which should be available by the time this program is released, is planned to have the ability to model the full L network feed system. This will allow evaluating array performance as frequency and component values are changed. Check with <http://eznec.com> for more information about EZNEC v. 5.0, and <http://eznec.com/misc/Arrayfeed/> for updates to this program, including a new manual which includes information about modeling the L network system with EZNEC v. 5.0. This completes the L network system design example.

Support

Program Support

This program is being provided free of charge, and no support is available. I would, however, appreciate reports of any bugs which are found, or errors or omissions in the manual. Please send reports to w7el@eznec.com. Corrections and updates to the program and manual, when available, will be posted at <http://eznec.com/misc/Arrayfeed/>.

Legal Notices

This program is copyright © 2007 by Roy W. Lewallen. It may be freely copied and distributed, providing no fee is charged. The user assumes all responsibility for its use. No warrantee of any kind is given or implied.