



Electrical Engineering Department,  
Fourth Year - Communications & Electronics.

# EE 466 ANTENNA

## Lab Assignment-2 (GUI)

---

PREPARED BY	SECTION	SEAT.NO.
Mahmoud Mohamed Kamal Ismail	7	250
Mahmoud AbdElHady Mahmoud	7	248
Ahmed Mohamed Abdelakher	1	37

## ➤ Contents

<b>1. Interface design &amp; Code</b>	<b>Page 4</b>
1.1. Select	Page 4
1.2. Sketch	Page 5
1.3. Reset	Page 10
1.4. Special Cases	Page 11
<b>2. Examples</b>	<b>Page 13</b>
2.1. Part 1: linear antenna (dipole of general length)	Page 13
2.2. Part 2: Uniform linear antenna array (ULA)	Page 15
2.3. Part 3: Nonuniformly-Fed linear antenna array	Page 16
A. Binomial Arrays	Page 16
B. Dolph-Tschebysceff Arrays	Page 17
2.4. Special Case for dipole antenna	Page
A. Infinite small dipole	
B. Short dipole	
C. $\lambda/2$ dipole	

## ➤ List of Figures

Figure 1: Interface design - linear antenna (dipole of general length).	Page 4
Figure 2: Select.	Page 4
Figure 3: Interface design - Uniform linear antenna array (ULA).	Page 5
Figure 4: Interface design - Nonuniformly-Fed linear antenna array (Binomial Arrays).	Page 5
Figure 5: Interface design - Nonuniformly-Fed linear antenna array (Dolph-Tschebysceff Arrays).	Page 6
Figure 6: Code – Select.	Page 6
Figure 7: Sketch Button.	Page 7
Figure 8: Code - Sketch.	Page 7 : Page 10
Figure 9: Reset.	Page 10
Figure 10: Code - Reset.	Page 10
Figure 11: Special Cases.	Page 11
Figure 12: Code - Special Cases.	Page 11 : Page 12
Figure 13: Example 1 - Part 1: linear antenna (dipole of general length).	Page 13
Figure 14: Example 2 (Errors) - Part 1: linear antenna (dipole of general length).	Page 14
Figure 15: Example 1 - Part 2: Uniform linear antenna array (ULA).	Page 15

Figure 16: Example 2 (Errors) - Part 2: Uniform linear antenna array (ULA).	Page 16
Figure 17: Example 1 - Part 3: Nonuniformly-Fed linear antenna array (Binomial Arrays).	Page 16
Figure 18: Example 2 (Errors) - Part 3: Nonuniformly-Fed linear antenna array (Binomial Arrays).	Page 17
Figure 19: Example 1 - Part 3: Nonuniformly-Fed linear antenna array (Dolph-Tschebysceff Arrays).	Page 17
Figure 20: Example 2 (Errors) - Part 3: Nonuniformly-Fed linear antenna array (Dolph-Tschebysceff Arrays).	Page 18
Figure 21: Infinite small dipole.	Page 19
Figure 22: Short dipole.	Page 19
Figure 23: $\lambda/2$ dipole.	Page 20

[https://github.com/MahmoudFierro98/Antenna\\_Lab/](https://github.com/MahmoudFierro98/Antenna_Lab/)

# 1. Interface design & Code

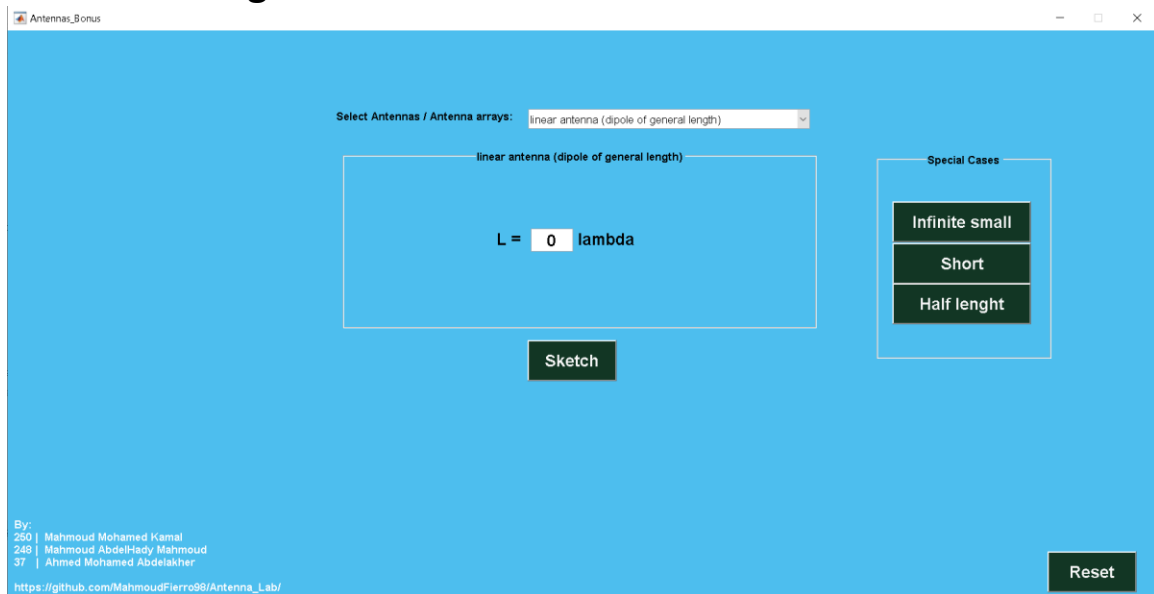


Figure 1: Interface design - linear antenna (dipole of general length).

## 1.1. Select

To choose any type of antennas you want.

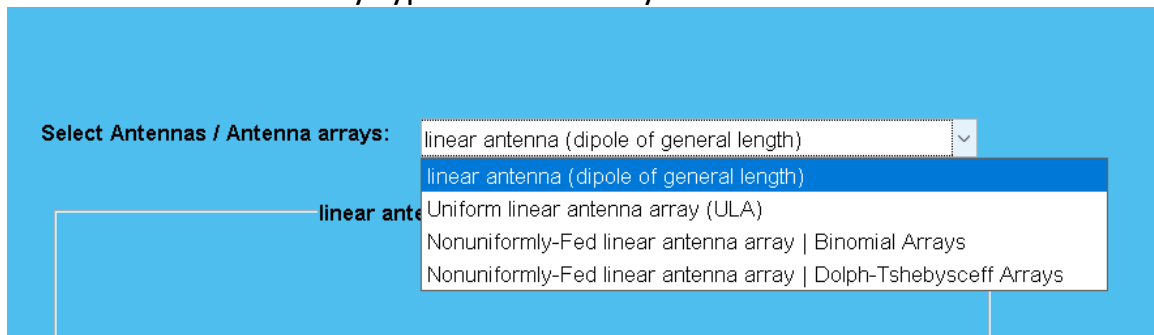


Figure 2: Select.

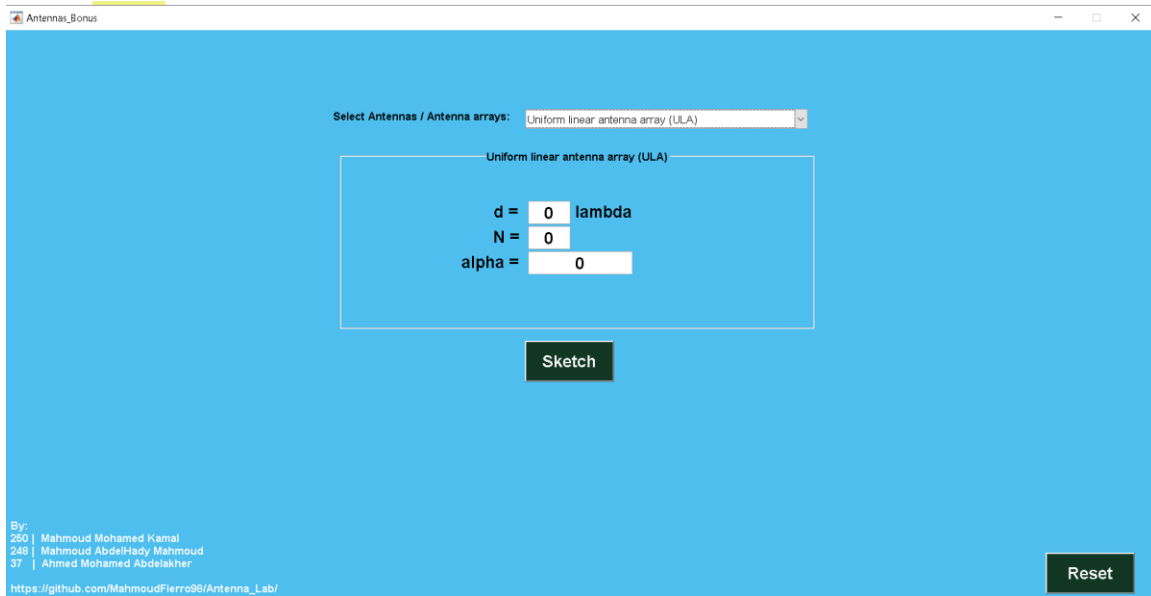


Figure 3: Interface design - Uniform linear antenna array (ULA).

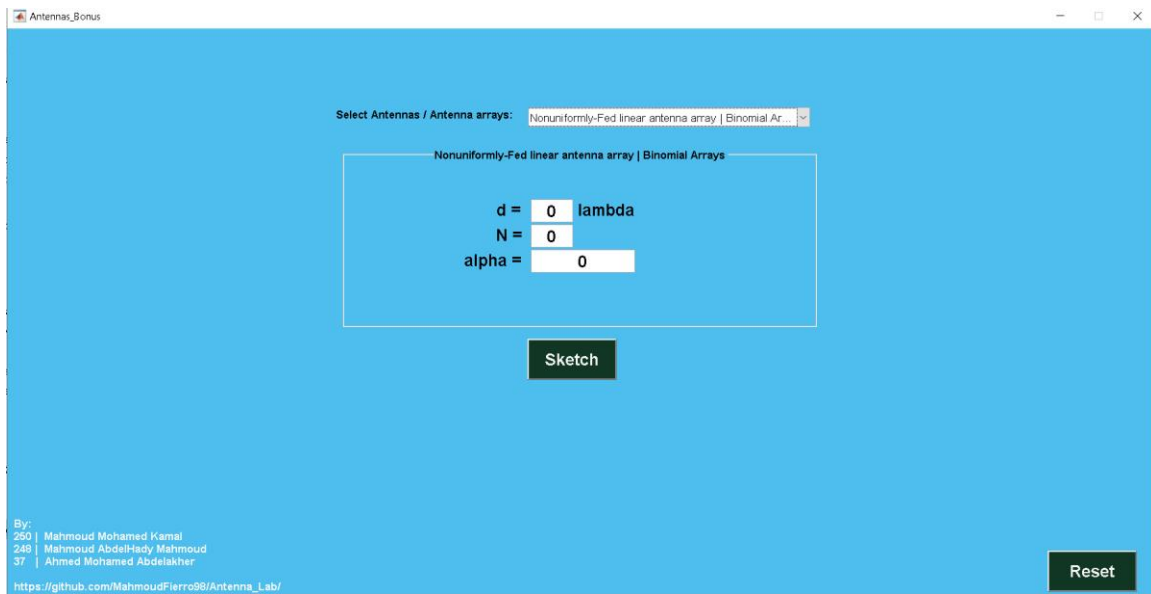


Figure 4: Interface design - Nonuniformly-Fed linear antenna array (Binomial Arrays).

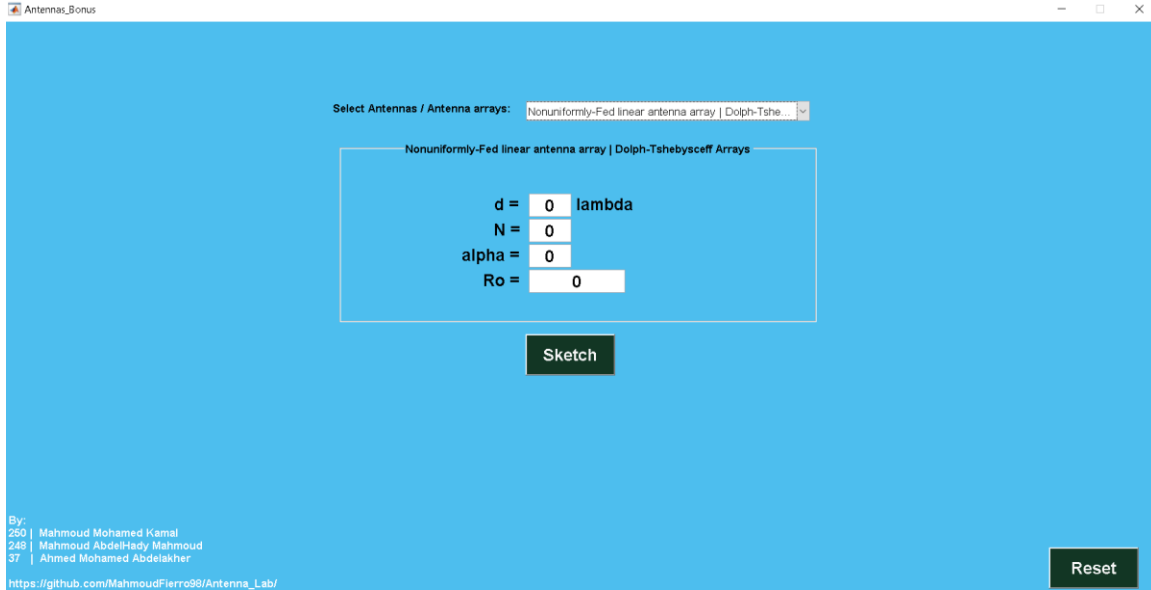


Figure 5: Interface design - Nonuniformly-Fed linear antenna array (Dolph-Tschebyscheff Arrays).

```

99 % --- Executes on selection change in select.
100 function select_Callback(hObject, eventdata, handles)
101     select = get(handles.select, 'Value');
102     set(handles.part1, 'Visible', 'off');
103     set(handles.part2, 'Visible', 'off');
104     set(handles.part3a, 'Visible', 'off');
105     set(handles.part3b, 'Visible', 'off');
106     set(handles.special_case, 'Visible', 'off');
107     set(handles.s1p, 'Visible', 'off');
108     set(handles.s2p, 'Visible', 'off');
109     set(handles.s3p, 'Visible', 'off');
110     set(handles.str, 'Visible', 'off');
111     set(handles.l_1, 'String', '0');
112     set(handles.d_2, 'String', '0');
113     set(handles.N_2, 'String', '0');
114     set(handles.alpha_2, 'String', '0');
115     set(handles.d_3a, 'String', '0');
116     set(handles.N_3a, 'String', '0');
117     set(handles.alpha_3a, 'String', '0');
118     set(handles.d_3b, 'String', '0');
119     set(handles.N_3b, 'String', '0');
120     set(handles.alpha_3b, 'String', '0');
121     set(handles.Ro_3b, 'String', '0');
122     switch select
123     case 1
124         set(handles.part1, 'Visible', 'on');
125         set(handles.special_case, 'Visible', 'on');
126     case 2
127         set(handles.part2, 'Visible', 'on');
128     case 3
129         set(handles.part3a, 'Visible', 'on');
130     case 4
131         set(handles.part3b, 'Visible', 'on');
132     end

```

Figure 6: Code – Select.

## 1.2. Sketch

### Sketch

Figure 7: Sketch Button.

```
346 % --- Executes on button press in sketch_button.
347 function sketch_button_Callback(hObject, eventdata, handles)
348     Lambda = 1;
349     B = (2*pi)/Lambda;
350     select = get(handles.select, 'Value');
351     set(handles.special_case, 'Visible', 'off');
352     set(handles.s1p, 'Visible', 'off');
353     set(handles.s2p, 'Visible', 'off');
354     set(handles.s3p, 'Visible', 'off');
355     set(handles.str, 'Visible', 'off');
356     figure(1);
357     figure(2);
358     figure(3);
359     switch select
360     case 1
361         set(handles.special_case, 'Visible', 'on');
362         l_1 = get(handles.l_1, 'String');
363         Theta = linspace(-pi, pi, 350);
364         Phi = linspace(-2*pi, 2*pi, 350);
365         L = str2num(l_1) * Lambda;
366         En = abs((cos((B*L)/2) * cos(Theta)) - cos((B*L)/2)) ./ sin(Theta));
367         if (L < 0)
368             set(handles.str, 'Visible', 'on');
369             set(handles.str, 'ForegroundColor', 'r');
370             set(handles.str, 'String', 'Error :: (L >= 0)');
371             close.figure(1);
372             close.figure(2);
373             close.figure(3);
374         else
375             set(handles.str, 'Visible', 'on');
376             set(handles.str, 'ForegroundColor', 'g');
377             set(handles.str, 'String', 'Done');
378             figure(1);
379             polar(Theta, En);
380             view([90 90]);
381             title('linear antenna (dipole of general length) - The 2D pattern of the dipole');
382             Phi_3D = meshgrid(Phi);
383             Theta_3D = meshgrid(Theta);
384             En_3D = meshgrid(En);
385             X = En_3D.*sin(Theta_3D).*cos(Phi_3D);
386             Y = En_3D.*sin(Theta_3D).*sin(Phi_3D);
387             Z = En_3D.*cos(Theta_3D);
388             figure(2);
389             surf(X, Y, Z, 'EdgeColor', 'interp', 'FaceAlpha', 0.1);
390             axis vis3d;
391             axis equal;
392             lighting gouraud;
393             title('linear antenna (dipole of general length) - The 3D pattern of the dipole');
394             close.figure(3);
395         end
396     case 2
397         d_2 = get(handles.d_2, 'String');
398         N_2 = get(handles.N_2, 'String');
399         alpha_2 = get(handles.alpha_2, 'String');
400         d = str2num(d_2) * Lambda;
401         N = str2num(N_2);
402         alpha = str2num(alpha_2);
403         Gamma = linspace(-pi, pi, 6000);
404         Phi = linspace(-pi, pi, 6000);
405         ebsi = B*d*cos(Gamma) + alpha;
406         AF = abs(sin((N*ebsi)/2) ./ (N * sin(ebsi/2)));
407         if (d < 0)
408             set(handles.str, 'Visible', 'on');
409             set(handles.str, 'ForegroundColor', 'r');
410             set(handles.str, 'String', 'Error :: (d >= 0)');
411             close.figure(1);
```



```

412 -         close(ffigure(2));
413 -         close(ffigure(3));
414 -     elseif (N < 0)
415 -         set(handles.str, 'Visible','on');
416 -         set(handles.str, 'ForegroundColor','r');
417 -         set(handles.str, 'String', 'Error :: (N >= 0)');
418 -         close(ffigure(1));
419 -         close(ffigure(2));
420 -         close(ffigure(3));
421 -     else
422 -         set(handles.str, 'Visible','on');
423 -         set(handles.str, 'ForegroundColor','g');
424 -         set(handles.str, 'String','Done');
425 -         figure(1);
426 -         plot(ebsi,AF);
427 -         title('Uniform linear antenna array (ULA) - array factor vs ebsi');
428 -         xlabel('ebsi','fontsize',10);
429 -         ylabel('AF','fontsize',10);
430 -         figure(2);
431 -         polar(Gamma,AF);
432 -         view([90 90]);
433 -         title('Uniform linear antenna array (ULA) - The 2D pattern of the array');
434 -         Phi_3D = meshgrid(Phi);
435 -         Gamma_3D = meshgrid(Gamma);
436 -         AF_3D = meshgrid(AF);
437 -         X = AF_3D.*sin(Gamma_3D).*cos(Phi_3D');
438 -         Y = AF_3D.*sin(Gamma_3D).*sin(Phi_3D');
439 -         Z = AF_3D.*cos(Gamma_3D);
440 -         figure(3);
441 -         surf(X,Y,Z,'EdgeColor','interp','FaceAlpha',0.1);
442 -         lighting gouraud;
443 -         title('Uniform linear antenna array (ULA) - The 3D pattern of the array');
444 -     end
445 - case 3
446 -     d_3a = get(handles.d_3a, 'String');
447 -     N_3a = get(handles.N_3a, 'String');
448 -     alpha_3a = get(handles.alpha_3a, 'String');
449 -     d = str2num(d_3a) * Lambda;
450 -     N = str2num(N_3a);
451 -     alpha = str2num(alpha_3a);
452 -     Theta = linspace(-pi,pi,6000);
453 -     Phi = linspace(-pi,pi,6000);
454 -     u = (B*d*cos(Theta) + alpha)/2;
455 -     AF = abs(cos(u).^(N-1));
456 -     if (d < 0)
457 -         set(handles.str, 'Visible','on');
458 -         set(handles.str, 'ForegroundColor','r');
459 -         set(handles.str, 'String', 'Error :: (d >= 0)');
460 -         close(ffigure(1));
461 -         close(ffigure(2));
462 -         close(ffigure(3));
463 -     elseif (N < 0)
464 -         set(handles.str, 'Visible','on');
465 -         set(handles.str, 'ForegroundColor','r');
466 -         set(handles.str, 'String', 'Error :: (N >= 0)');
467 -         close(ffigure(1));
468 -         close(ffigure(2));
469 -         close(ffigure(3));
470 -     else
471 -         set(handles.str, 'Visible','on');
472 -         set(handles.str, 'ForegroundColor','g');
473 -         set(handles.str, 'String','Done');
474 -         figure(1);
475 -         plot(u,AF);
476 -         title('Nonuniformly-Fed linear antenna array | Binomial Arrays - array factor vs u');
477 -         xlabel('u','fontsize',10);
478 -         ylabel('AF','fontsize',10);

```

```

478 -         ylabel('AF','fontsize',10);
479 -         figure(2);
480 -         polar(Theta,AF);
481 -         view([90 90]);
482 -         title('Nonuniformly-Fed linear antenna array | Binomial Arrays - The 2D pattern of the array');
483 -         Phi_3D = meshgrid(Phi);
484 -         Theta_3D = meshgrid(Theta);
485 -         AF_3D = meshgrid(AF);
486 -         X = AF_3D.*sin(Theta_3D).*cos(Phi_3D');
487 -         Y = AF_3D.*sin(Theta_3D).*sin(Phi_3D');
488 -         Z = AF_3D.*cos(Theta_3D);
489 -         figure(3);
490 -         surf(X,Y,Z,'EdgeColor','interp','FaceAlpha',0.1);
491 -         axis vis3d;
492 -         axis equal;
493 -         lighting gouraud;
494 -         title('Nonuniformly-Fed linear antenna array | Binomial Arrays - The 3D pattern of the array');
495 -     end
496 - case 4
497 -     d_3b = get(handles.d_3b, 'String');
498 -     N_3b = get(handles.N_3b, 'String');
499 -     alpha_3b = get(handles.alpha_3b, 'String');
500 -     Ro_3b = get(handles.Ro_3b, 'String');
501 -     d = str2num(d_3b) * Lambda;
502 -     N = str2num(N_3b);
503 -     alpha = str2num(alpha_3b);
504 -     Ro = str2num(Ro_3b);
505 -     M = N - 1;
506 -     Zo = cosh((1/M)*acosh(Ro));
507 -     Z = linspace(-Zo,Zo,6000);
508 -     u_up = acos(Z./Zo);
509 -     u_down = -u_up;
510 -     u = [u_down ; u_up];
511 -     Theta1 = acos(((2.*u_down)-alpha)/(B*d));
512 -     Theta2 = -Theta1;
513 -     Phi = linspace(-pi,pi,6000);
514 -     AF = abs(cosh(M.*acosh(Z)));
515 -     if (d < 0)
516 -         set(handles.str, 'Visible','on');
517 -         set(handles.str, 'ForegroundColor','r');
518 -         set(handles.str, 'String', 'Error :: (d >= 0)');
519 -         close(figure(1));
520 -         close(figure(2));
521 -         close(figure(3));
522 -     elseif (N < 0)
523 -         set(handles.str, 'Visible','on');
524 -         set(handles.str, 'ForegroundColor','r');
525 -         set(handles.str, 'String', 'Error :: (N >= 0)');
526 -         close(figure(1));
527 -         close(figure(2));
528 -         close(figure(3));
529 -     elseif (Ro <= 1)
530 -         set(handles.str, 'Visible','on');
531 -         set(handles.str, 'ForegroundColor','r');
532 -         set(handles.str, 'String', 'Error :: always (Ro > 1)');
533 -         close(figure(1));
534 -         close(figure(2));
535 -         close(figure(3));
536 -     else
537 -         set(handles.str, 'Visible','on');
538 -         set(handles.str, 'ForegroundColor','g');
539 -         set(handles.str, 'String','Done');
540 -         figure(1);
541 -         plot(Z,AF);
542 -         title('Nonuniformly-Fed linear antenna array | Dolph-Tshebyscheff Arrays - array factor vs Z');
543 -         xlabel('Z','fontsize',10);

```

```

533 -         close(fgure(1));
534 -         close(fgure(2));
535 -         close(fgure(3));
536 -     else
537 -         set(handles.ctr, 'Visible','on');
538 -         set(handles.ctr, 'ForegroundColor','g');
539 -         set(handles.ctr, 'String','Done');
540 -         fgure(1);
541 -         plot(Z,AF);
542 -         title('Nonuniformly-Fed linear antenna array | Dolph-Tshebysceff Arrays - array factor vs Z');
543 -         xlabel('Z','fontsize',10);
544 -         ylabel('AF','fontsize',10);
545 -         fgure(2);
546 -         polar(Theta1,AF,'-b');
547 -         hold on;
548 -         polar(Theta2,AF,'-b');
549 -         view([90 90]);
550 -         title('Nonuniformly-Fed linear antenna array | Dolph-Tshebysceff Arrays - The 2D pattern of the array');
551 -         Phi_3D = meshgrid(Phi);
552 -         Theta_3D = meshgrid(Theta1);
553 -         AF_3D = meshgrid(AF);
554 -         X = AF_3D.*sin(Theta_3D).*cos(Phi_3D');
555 -         Y = AF_3D.*sin(Theta_3D).*sin(Phi_3D');
556 -         Z = AF_3D.*cos(Theta_3D);
557 -         fgure(3);
558 -         surf(X,Y,Z,'EdgeColor','interp','FaceAlpha',0.1);
559 -         axis vis3d;
560 -         axis equal;
561 -         lighting gouraud;
562 -         title('Nonuniformly-Fed linear antenna array | Dolph-Tshebysceff Arrays - The 3D pattern of the array');
563 -     end
564 - end

```

Figure 8: Code - Sketch.

### 1.3. Reset

To reboot program and close all fgures.



Reset

Figure 9: Reset.

```

566 - % --- Executes on button press in reset_button.
567 - function reset_button_Callback(hObject, eventdata, handles)
568 - set(handles.select, 'Value',1);
569 - set(handles.part1, 'Visible','on');
570 - set(handles.part2, 'Visible','off');
571 - set(handles.part3a, 'Visible','off');
572 - set(handles.part3b, 'Visible','off');
573 - set(handles.special_case, 'Visible','on');
574 - set(handles.s1p, 'Visible','off');
575 - set(handles.s2p, 'Visible','off');
576 - set(handles.s3p, 'Visible','off');
577 - set(handles.ctr, 'Visible','off');
578 - set(handles.l_1, 'String','0');
579 - set(handles.d_2, 'String','0');
580 - set(handles.N_2, 'String','0');
581 - set(handles.alpha_2, 'String','0');
582 - set(handles.d_3a, 'String','0');
583 - set(handles.N_3a, 'String','0');
584 - set(handles.alpha_3a, 'String','0');
585 - set(handles.d_3b, 'String','0');
586 - set(handles.N_3b, 'String','0');
587 - set(handles.alpha_3b, 'String','0');
588 - set(handles.Ro_3b, 'String','0');
589 - close(fgure(1));
590 - close(fgure(2));
591 - close(fgure(3));

```

Figure 10: Code - Reset.

## 1.4. Special Cases

Show only if you select linear antenna (dipole of general length).

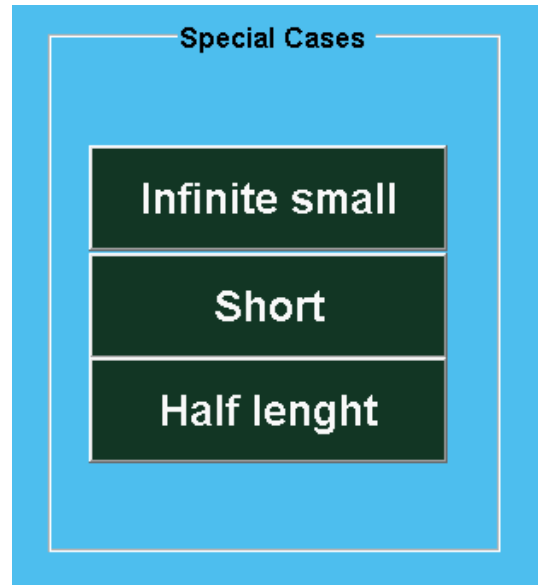


Figure 11: Special Cases.

```
594 % --- Executes on button press in s1.
595 function s1_Callback(hObject, eventdata, handles)
596     Lambda = 1;
597     B       = (2*pi)/Lambda;
598     set(handles.s1p, 'Visible', 'on');
599     set(handles.s2p, 'Visible', 'off');
600     set(handles.s3p, 'Visible', 'off');
601     set(handles.str, 'Visible', 'off');
602     Theta = linspace(-pi,pi,350);
603     Phi    = linspace(-2*pi,2*pi,350);
604     L      = (1/50) * Lambda;
605     En     = abs(sin(Theta));
606     set(handles.str, 'Visible', 'on');
607     set(handles.str, 'ForegroundColor', 'g');
608     set(handles.str, 'String', 'Done');
609     figure(1);
610     polar(Theta, En);
611     view([90 90]);
612     title('Infinte small dipole - The 2D pattern of the dipole');
613     Phi_3D = meshgrid(Phi);
614     Theta_3D = meshgrid(Theta);
615     En_3D = meshgrid(En);
616     X      = En_3D.*sin(Theta_3D).*cos(Phi_3D);
617     Y      = En_3D.*sin(Theta_3D).*sin(Phi_3D);
618     Z      = En_3D.*cos(Theta_3D);
619     figure(2);
620     surf(X,Y,Z, 'EdgeColor', 'interp', 'FaceAlpha', 0.1);
621     axis vis3d;
622     axis equal;
623     lighting gouraud;
624     title('Infinte small dipole - The 3D pattern of the dipole');
625     close.figure(3);
```

```

627 % --- Executes on button press in s2.
628 function s2_Callback(hObject, eventdata, handles)
629 Lambda = 1;
630 B = (2*pi)/Lambda;
631 set(handles.s2p, 'Visible', 'on');
632 set(handles.s1p, 'Visible', 'off');
633 set(handles.s3p, 'Visible', 'off');
634 set(handles.str, 'Visible', 'off');
635 Theta = linspace(-pi,pi,350);
636 Phi = linspace(-2*pi,2*pi,350);
637 L = (1/10) * Lambda;
638 En = abs(sin(Theta));
639 set(handles.str, 'Visible', 'on');
640 set(handles.str, 'ForegroundColor', 'g');
641 set(handles.str, 'String', 'Done');
642 figure(1);
643 polar(Theta,En);
644 view([90 90]);
645 title('Short dipole - The 2D pattern of the dipole');
646 Phi_3D = meshgrid(Phi);
647 Theta_3D = meshgrid(Theta);
648 En_3D = meshgrid(En);
649 X = En_3D.*sin(Theta_3D).*cos(Phi_3D');
650 Y = En_3D.*sin(Theta_3D).*sin(Phi_3D');
651 Z = En_3D.*cos(Theta_3D);
652 figure(2);
653 surf(X,Y,Z,'EdgeColor','interp','FaceAlpha',0.1);
654 axis vis3d;
655 axis equal;
656 lighting gouraud;
657 title('Short dipole - The 3D pattern of the dipole');
658 close.figure(3));
659
660 % --- Executes on button press in s3.
661 function s3_Callback(hObject, eventdata, handles)
662 Lambda = 1;
663 B = (2*pi)/Lambda;
664 set(handles.s3p, 'Visible', 'on');
665 set(handles.s1p, 'Visible', 'off');
666 set(handles.s2p, 'Visible', 'off');
667 set(handles.str, 'Visible', 'off');
668 Theta = linspace(-pi,pi,350);
669 Phi = linspace(-2*pi,2*pi,350);
670 L = (1/2) * Lambda;
671 En = abs((cos((B*L)/2).*cos(Theta)) - cos((B*L)/2)) ./ sin(Theta);
672 set(handles.str, 'Visible', 'on');
673 set(handles.str, 'ForegroundColor', 'g');
674 set(handles.str, 'String', 'Done');
675 figure(1);
676 polar(Theta,En);
677 view([90 90]);
678 title('lambda/2 dipole - The 2D pattern of the dipole');
679 Phi_3D = meshgrid(Phi);
680 Theta_3D = meshgrid(Theta);
681 En_3D = meshgrid(En);
682 X = En_3D.*sin(Theta_3D).*cos(Phi_3D');
683 Y = En_3D.*sin(Theta_3D).*sin(Phi_3D');
684 Z = En_3D.*cos(Theta_3D);
685 figure(2);
686 surf(X,Y,Z,'EdgeColor','interp','FaceAlpha',0.1);
687 axis vis3d;
688 axis equal;
689 lighting gouraud;
690 title('lambda/2 dipole - The 3D pattern of the dipole');
691 close.figure(3));

```

Figure 12: Code - Special Cases.

## 2. Examples

### 2.1. Part 1: linear antenna (dipole of general length)

Example 1:  $l = 2\lambda$

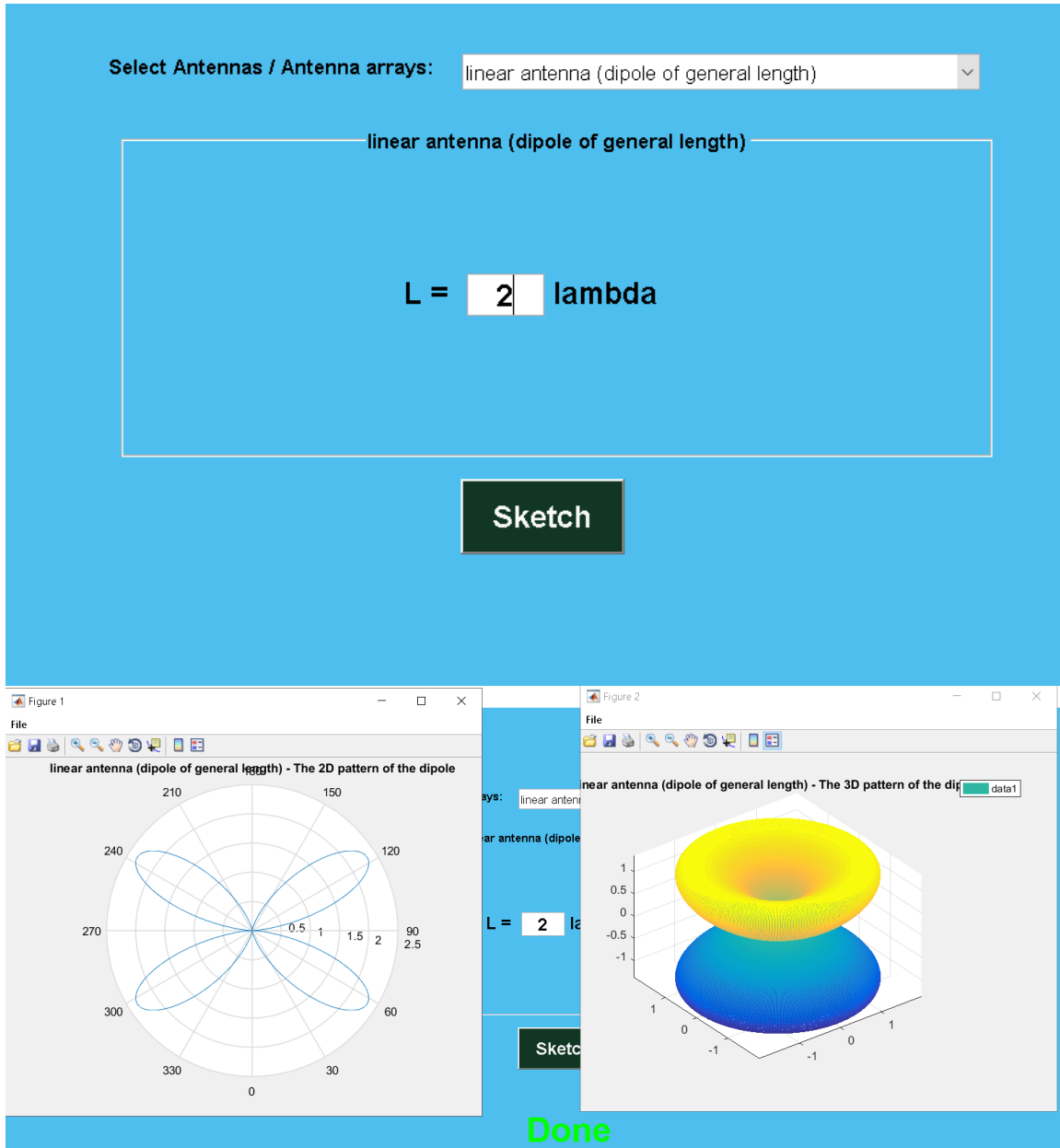



Figure 13: Example 1 - Part 1: linear antenna (dipole of general length).

Example 2: *Errors*  
if  $l < 0$

Select Antennas / Antenna arrays: linear antenna (dipole of general length) 

linear antenna (dipole of general length)

$L = -1 \text{ lambda}$

**Sketch**

**Error :: ( $L \geq 0$ )**

Figure 14: Example 2 (Errors) - Part 1: linear antenna (dipole of general length).

## 2.2. Part 2: Uniform linear antenna array (ULA)

Example 1:  $d = \frac{4\lambda}{7}$ ,  $N = 7$ ,  $\alpha = \frac{-4\pi}{7}$

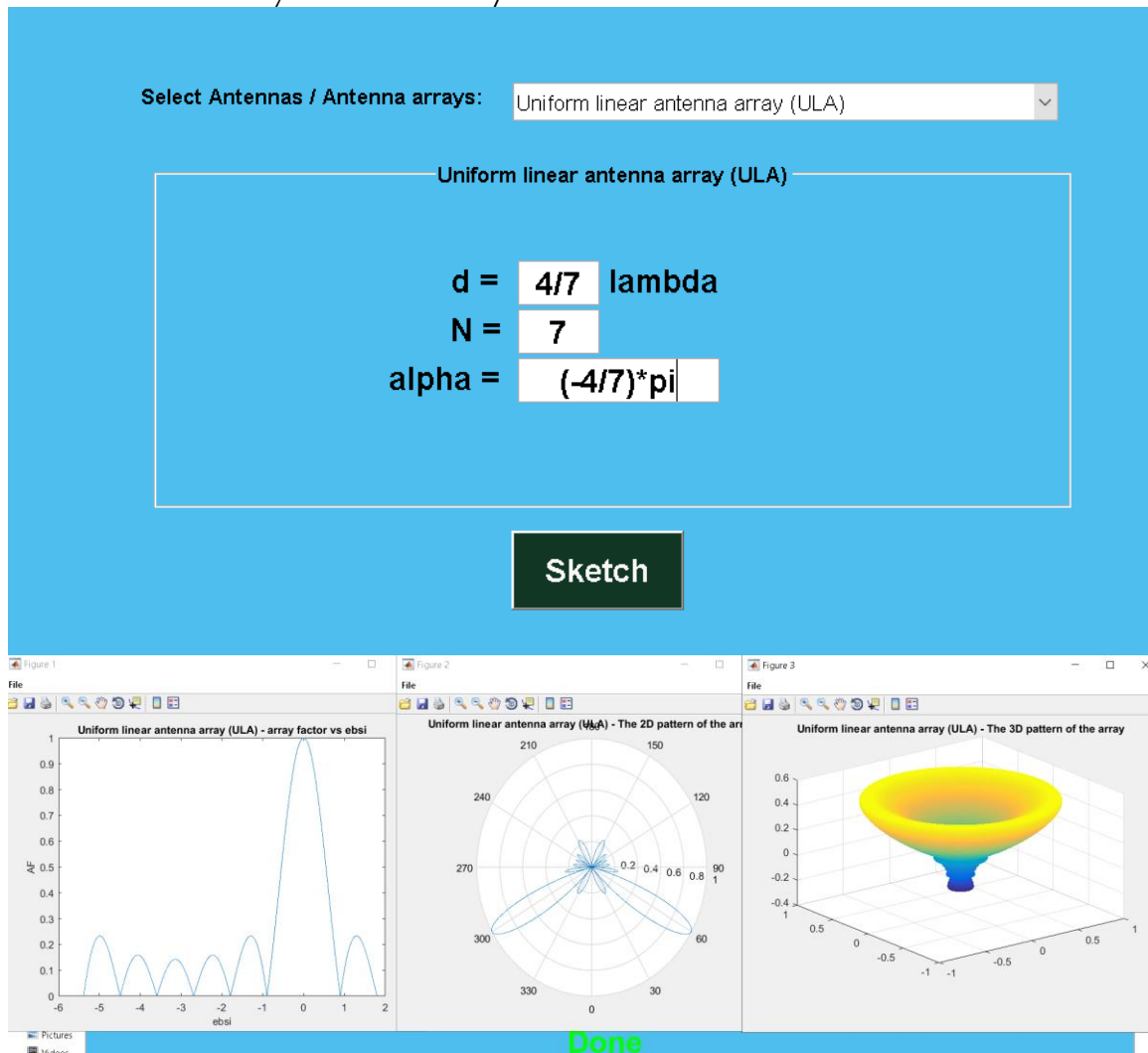


Figure 15: Example 1 - Part 2: Uniform linear antenna array (ULA).



Example 3: *Errors*  
if  $d < 0$  or  $N < 0$ .

Figure 16: Example 2 (Errors) - Part 2: Uniform linear antenna array (ULA).

## 2.3. Part 3: Nonuniformly-Fed linear antenna array

### A. Binomial Arrays

Example 1:  $d = \frac{3\lambda}{4}$ ,  $N = 8$ ,  $\alpha = 0$

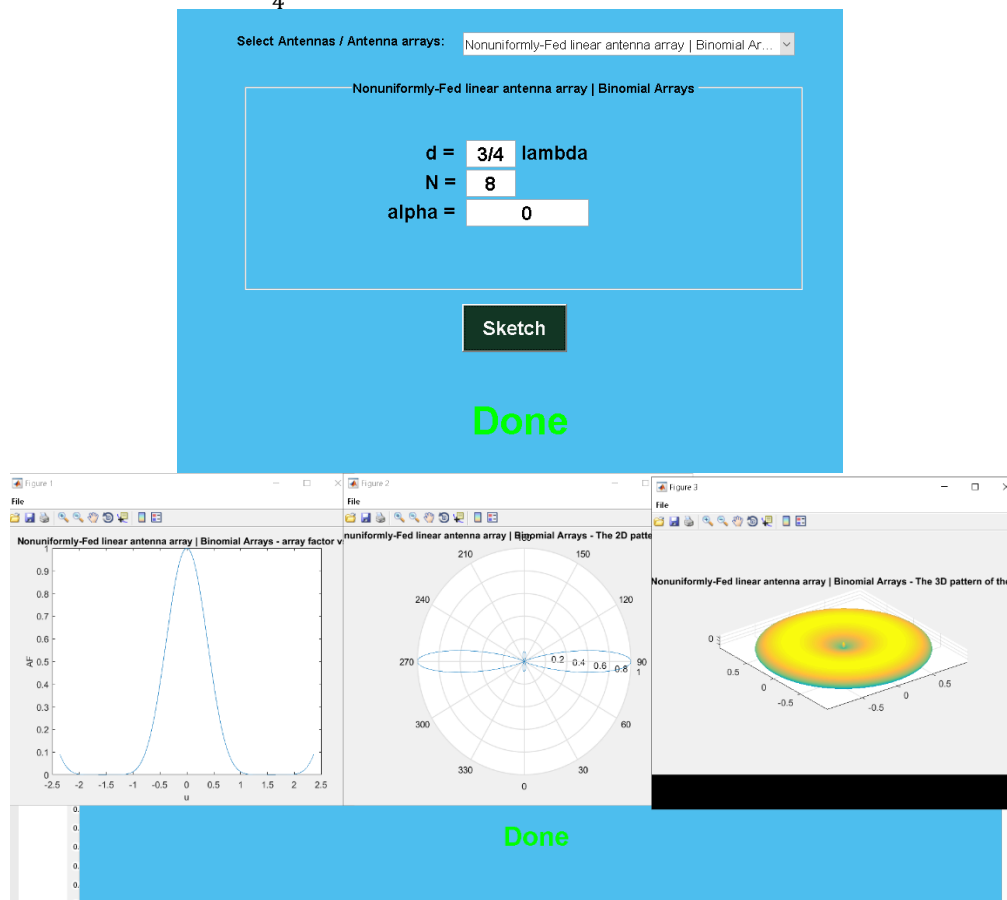


Figure 17: Example 1 - Part 3: Nonuniformly-Fed linear antenna array (Binomial Arrays).

Example 3: *Errors*  
if  $d < 0$  or  $N < 0$ .

Select Antennas / Antenna arrays: Nonuniformly-Fed linear antenna array | Binomial Ar...

Nonuniformly-Fed linear antenna array | Binomial Arrays

d = -5 lambda

N = 0

alpha = 0

Sketch

Error :: (d >= 0)

Select Antennas / Antenna arrays: Nonuniformly-Fed linear antenna array | Binomial Ar...

Nonuniformly-Fed linear antenna array | Binomial Arrays

d = 0 lambda

N = -2

alpha = 0

Sketch

Error :: (N >= 0)

Figure 18: Example 2 (Errors) - Part 3: Nonuniformly-Fed linear antenna array (Binomial Arrays).

## B. Dolph-Tschebyscheff Arrays

Example 1:  $d = \frac{\lambda}{2}$ ,  $N = 6$ ,  $\alpha = -\pi$ ,  $R_0 = 10$ .

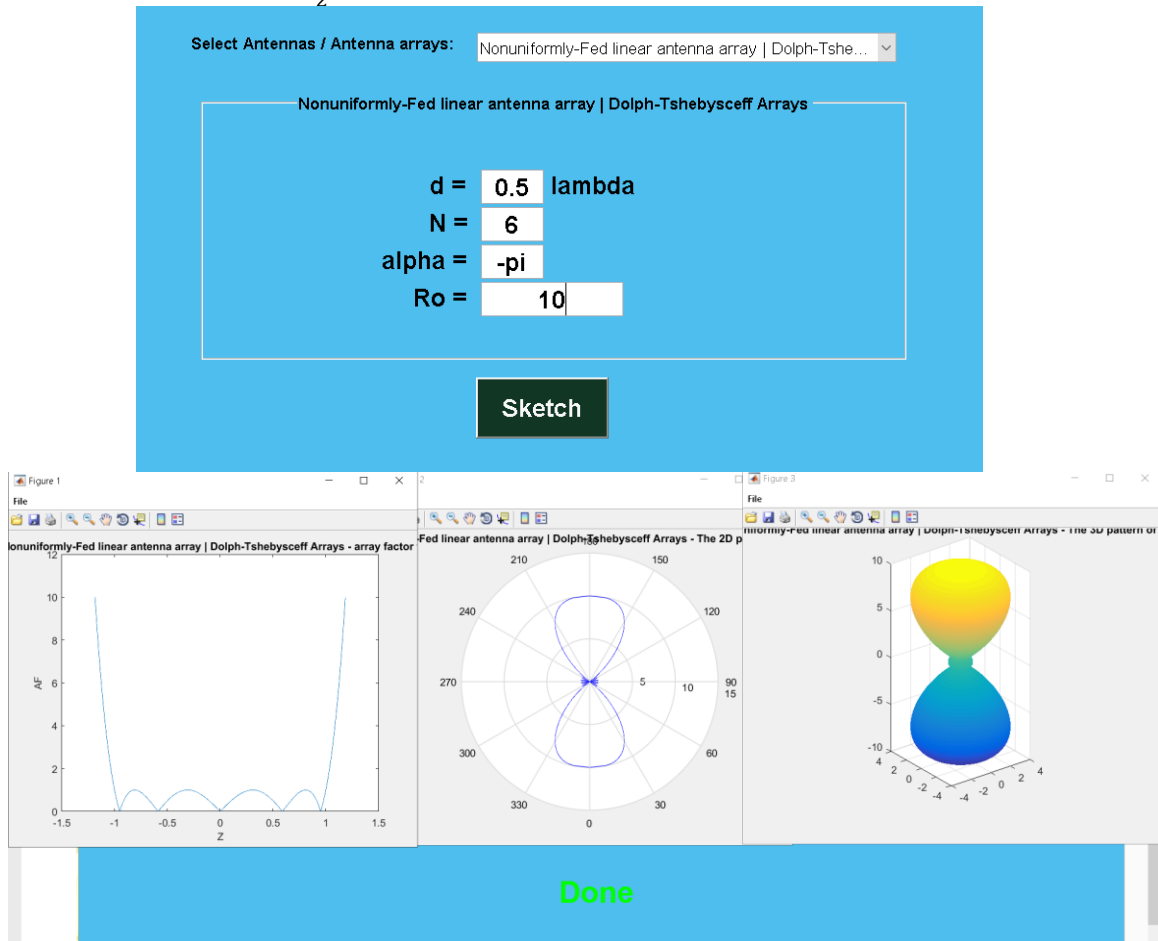


Figure 19: Example 1 - Part 3: Nonuniformly-Fed linear antenna array (Dolph-Tschebyscheff Arrays).

Example 3: *Errors*

if  $d < 0$  or  $N < 0$ , we will ask you again.

Always  $R_0 > 1$ .

Select Antennas / Antenna arrays: Nonuniformly-Fed linear antenna array | Dolph-Tshe... ▼

Nonuniformly-Fed linear antenna array | Dolph-Tshebysceff Arrays

d =  lambda

N =

alpha =

Ro =

**Sketch**

**Error :: always ( $R_0 > 1$ )**

Figure 20: Example 2 (Errors) - Part 3: Nonuniformly-Fed linear antenna array (Dolph-Tschebysceff Arrays).

## 2.4. Special Case for dipole antenna

### A. Infinite small dipole

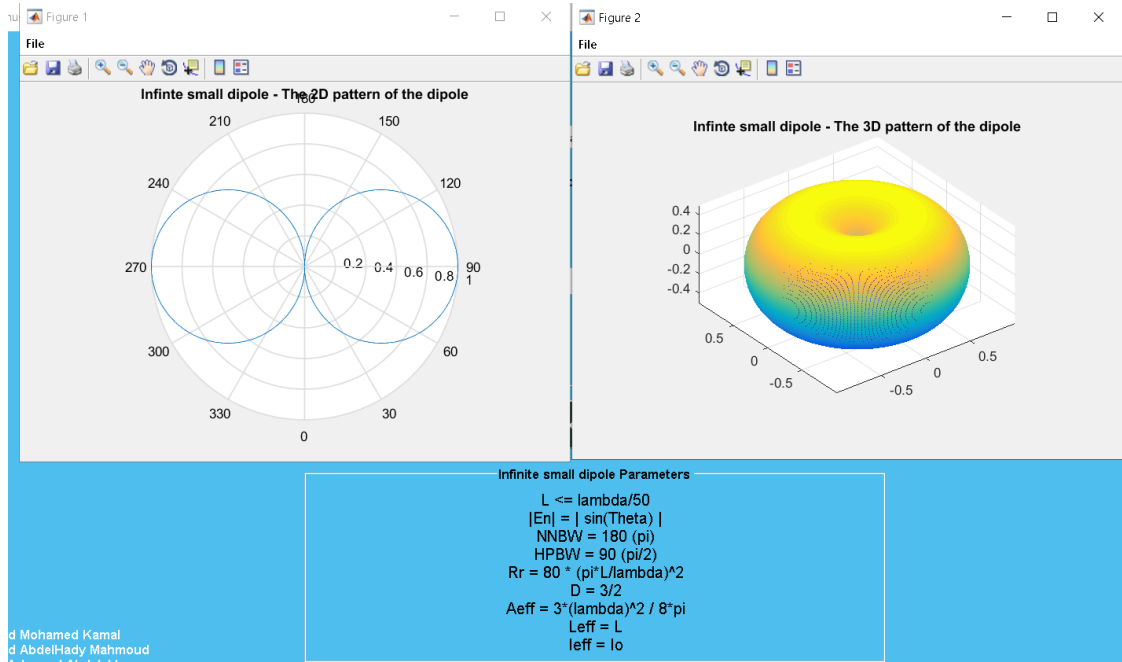


Figure 21: Infinite small dipole.

### B. Short dipole

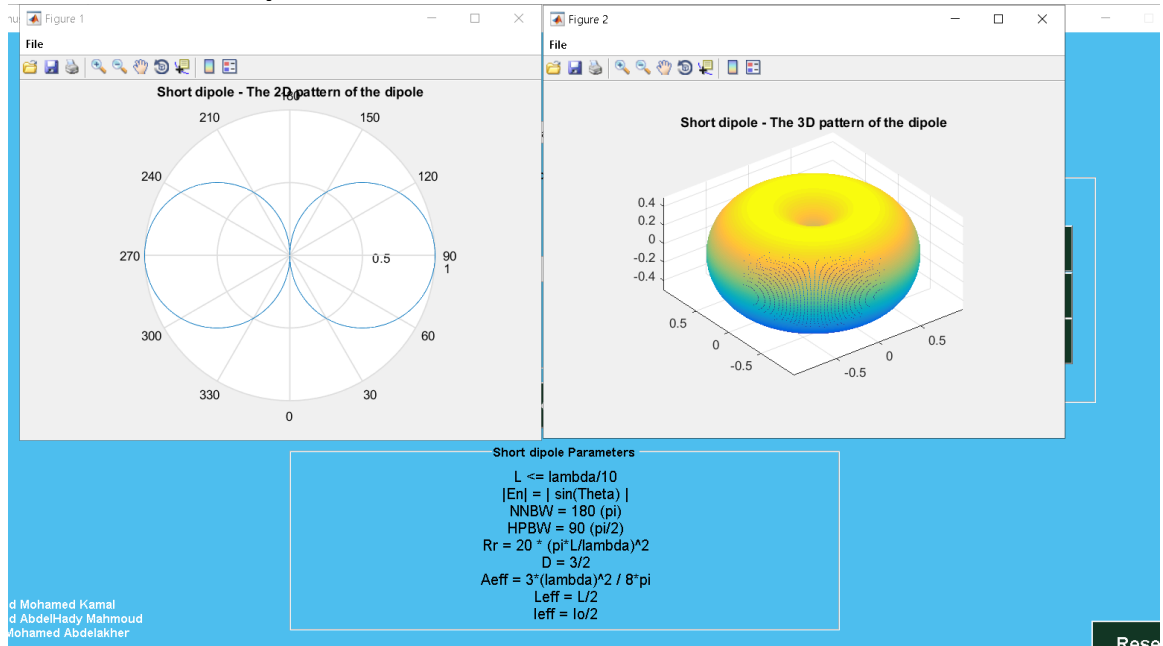


Figure 22: Short dipole.

## C. $\lambda/2$ dipole

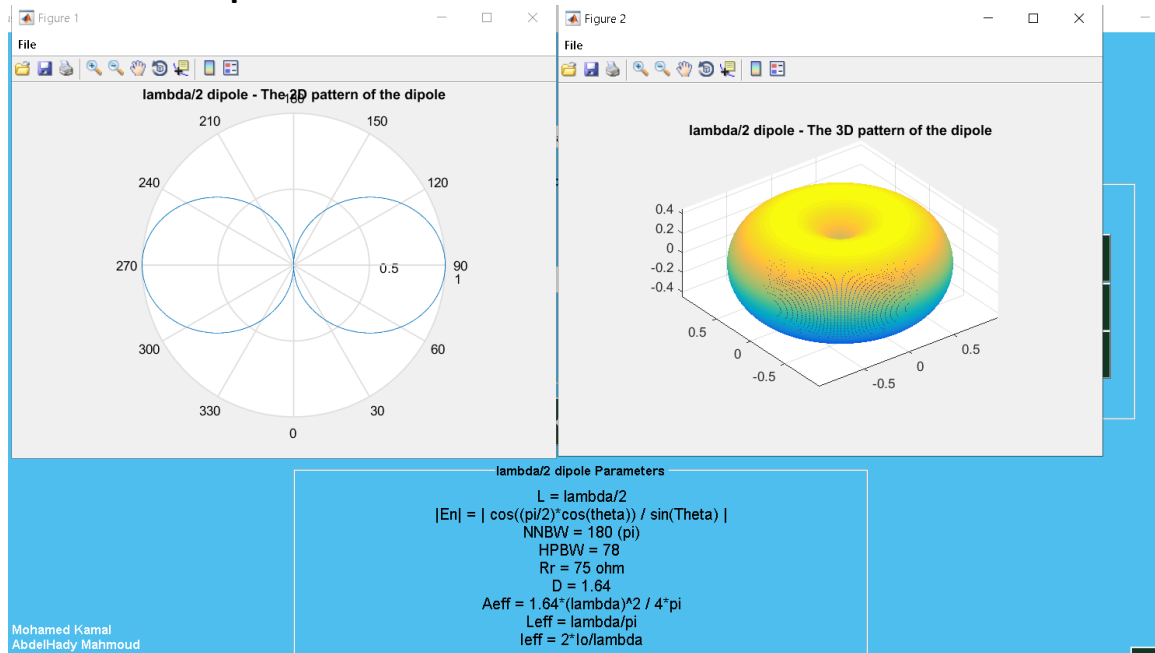


Figure 23:  $\lambda/2$  dipole.