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

➢ List of Figures

https://github.com/MahmoudFierro98/Antenna_Lab/

1. Interface design & Code

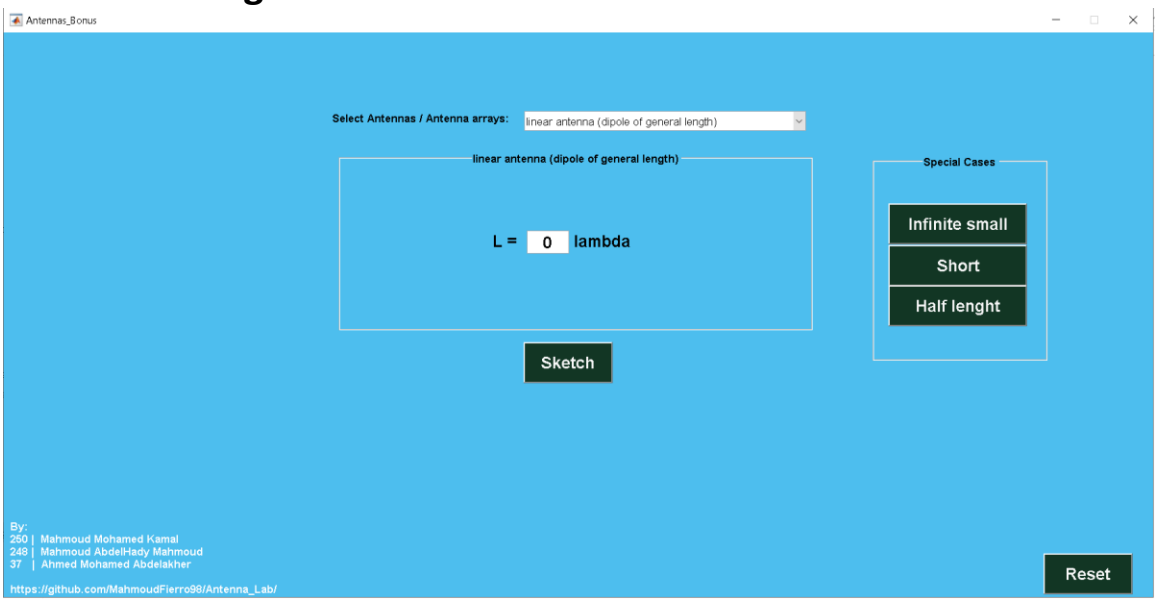


Figure 1: Interface design - linear antenna (dipole of general length).

1.1. Select

To choose any type of antennas you want.

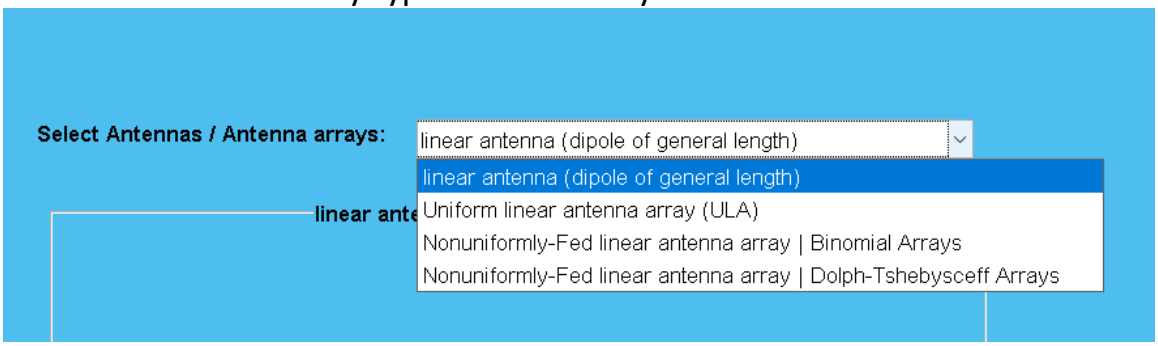


Figure 2: Select.

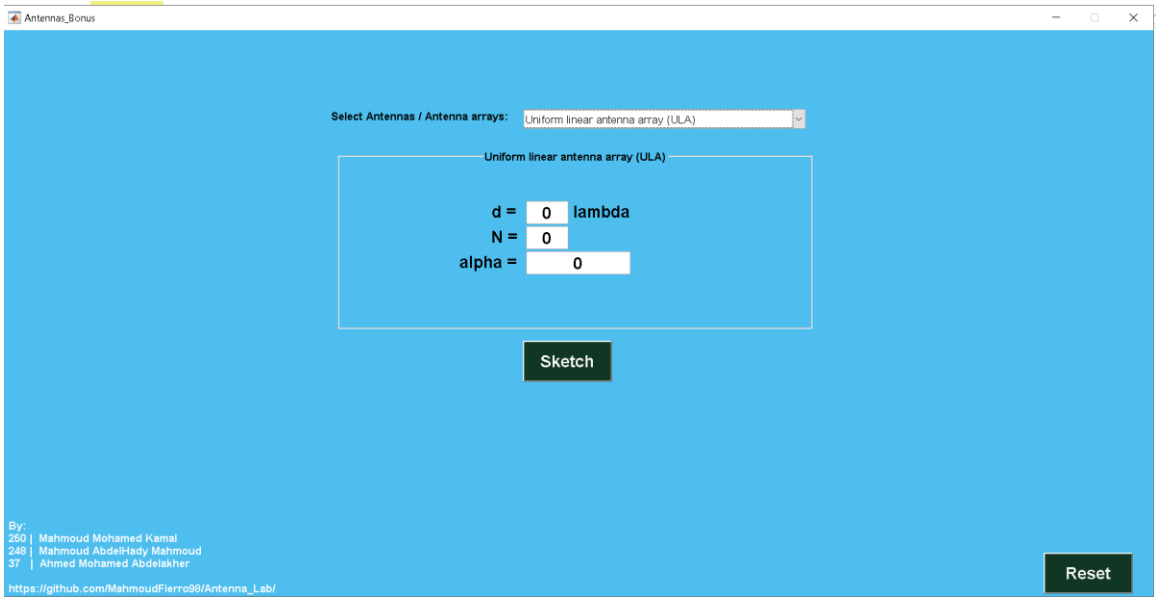


Figure 3: Interface design - Uniform linear antenna array (ULA).

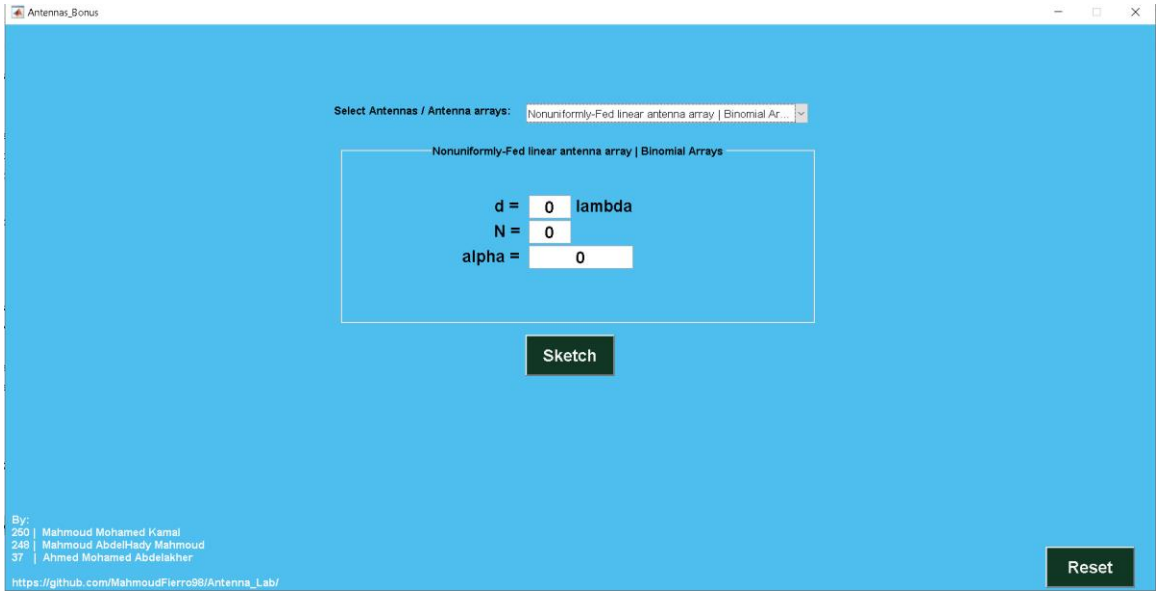


Figure 4: Interface design - Nonuniformly-Fed linear antenna array (Binomial Arrays).

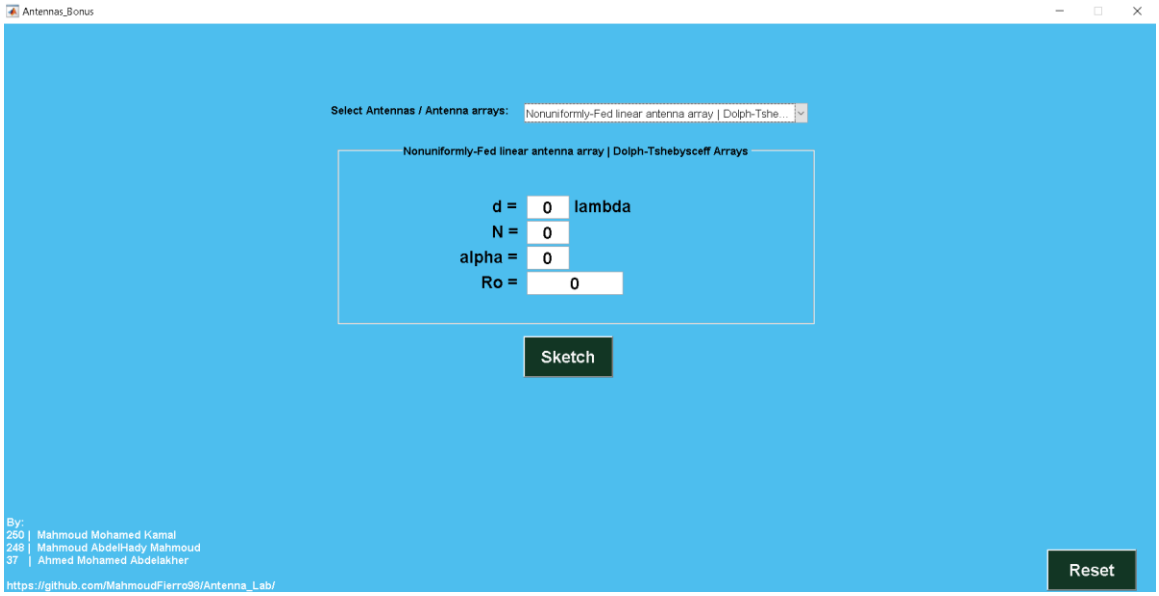


Figure 5: Interface design - Nonuniformly-Fed linear antenna array (Dolph-Tschebysceff Arrays).

```
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

Figure 7: Sketch Button.

```
346     % --- Executes on button press in sketch_buttom.
347   function sketch_buttom_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));
```

```
            close(figure(2));
            close(figure(3));
        elseif (N < 0)
            set(handles.str, 'Visible','on');
            set(handles.str, 'ForegroundColor','r');
            set(handles.str, 'String', 'Error :: (N >= 0)');
            close(figure(1));
            close(figure(2));
            close(figure(3));
        else
            set(handles.str, 'Visible','on');
            set(handles.str, 'ForegroundColor','g');
            set(handles.str, 'String','Done');
            figure(1);
            plot(ebsi,AF);
            title('Uniform linear antenna array (ULA) - array factor vs ebsi');
            xlabel('ebsi','fontsize',10);
            ylabel('AF','fontsize',10);
            figure(2);
            polar(Gamma,AF);
            view([90 90]);
            title('Uniform linear antenna array (ULA) - The 2D pattern of the array');
            Phi_3D   = meshgrid(Phi);
            Gamma_3D = meshgrid(Gamma);
            AF_3D    = meshgrid(AF);
            X        = AF_3D.*sin(Gamma_3D).*cos(Phi_3D');
            Y        = AF_3D.*sin(Gamma_3D).*sin(Phi_3D');
            Z        = AF_3D.*cos(Gamma_3D);
            figure(3);
            surf(X,Y,Z,'EdgeColor','interp','FaceAlpha',0.1);
            lighting gouraud;
            title('Uniform linear antenna array (ULA) - The 3D pattern of the array');
        end
    case 3
        d_3a     = get(handles.d_3a,     'String');
        N_3a     = get(handles.N_3a,     'String');
        alpha_3a = get(handles.alpha_3a, 'String');
        d        = str2num(d_3a) * Lambda;
        N        = str2num(N_3a);
        alpha    = str2num(alpha_3a);
        Theta    = linspace(-pi,pi,6000);
        Phi      = linspace(-pi,pi,6000);
        u        = (B*d*cos(Theta) + alpha)/2;
        AF       = abs(cos(u).^(N-1));
        if (d < 0)
            set(handles.str, 'Visible','on');
            set(handles.str, 'ForegroundColor','r');
            set(handles.str, 'String', 'Error :: (d >= 0)');
            close(figure(1));
            close(figure(2));
            close(figure(3));
        elseif (N < 0)
            set(handles.str, 'Visible','on');
            set(handles.str, 'ForegroundColor','r');
            set(handles.str, 'String', 'Error :: (N >= 0)');
            close(figure(1));
            close(figure(2));
            close(figure(3));
        else
            set(handles.str, 'Visible','on');
            set(handles.str, 'ForegroundColor','g');
            set(handles.str, 'String','Done');
            figure(1);
            plot(u,AF);
            title('Nonuniformly-Fed linear antenna array | Binomial Arrays - array factor vs u');
            xlabel('u','fontsize',10);
```

```
            ylabel('AF','fontsize',10);
            figure(2);
            polar(Theta,AF);
            view([90 90]);
            title('Nonuniformly-Fed linear antenna array | Binomial Arrays - The 2D pattern of the array');
            Phi_3D   = meshgrid(Phi);
            Theta_3D = meshgrid(Theta);
            AF_3D    = meshgrid(AF);
            X        = AF_3D.*sin(Theta_3D).*cos(Phi_3D');
            Y        = AF_3D.*sin(Theta_3D).*sin(Phi_3D');
            Z        = AF_3D.*cos(Theta_3D);
            figure(3);
            surf(X,Y,Z,'EdgeColor','interp','FaceAlpha',0.1);
            axis vis3d;
            axis equal;
            lighting gouraud;
            title('Nonuniformly-Fed linear antenna array | Binomial Arrays - The 3D pattern of the array');
        end
    case 4
        d_3b     = get(handles.d_3b,     'String');
        N_3b     = get(handles.N_3b,     'String');
        alpha_3b = get(handles.alpha_3b, 'String');
        Ro_3b    = get(handles.Ro_3b,    'String');
        d        = str2num(d_3b) * Lambda;
        N        = str2num(N_3b);
        alpha    = str2num(alpha_3b);
        Ro       = str2num(Ro_3b);
        M        = N - 1;
        Zo       = cosh((1/M)*acosh(Ro));
        Z        = linspace(-Zo,Zo,6000);
        u_up     = acos(Z./Zo);
        u_down   = -u_up;
        u        = [u_down ; u_up];
        Theta1   = acos(((2.*u_down)-alpha)/(B*d));
        Theta2   = -Theta1;
        Phi      = linspace(-pi,pi,6000);
        AF       = abs(cosh(M.*acosh(Z)));
        if (d < 0)
            set(handles.str, 'Visible','on');
            set(handles.str, 'ForegroundColor','r');
            set(handles.str, 'String', 'Error :: (d >= 0)');
            close(figure(1));
            close(figure(2));
            close(figure(3));
        elseif (N < 0)
            set(handles.str, 'Visible','on');
            set(handles.str, 'ForegroundColor','r');
            set(handles.str, 'String', 'Error :: (N >= 0)');
            close(figure(1));
            close(figure(2));
            close(figure(3));
        elseif (Ro <= 1)
            set(handles.str, 'Visible','on');
            set(handles.str, 'ForegroundColor','r');
            set(handles.str, 'String', 'Error :: always (Ro > 1)');
            close(figure(1));
            close(figure(2));
            close(figure(3));
        else
            set(handles.str, 'Visible','on');
            set(handles.str, 'ForegroundColor','g');
            set(handles.str, 'String','Done');
            figure(1);
            plot(Z,AF);
            title('Nonuniformly-Fed linear antenna array | Dolph-Tshebysceff Arrays - array factor vs Z');
            xlabel('Z','fontsize',10);
```

```
            close(figure(1));
            close(figure(2));
            close(figure(3));
        else
            set(handles.str, 'Visible','on');
            set(handles.str, 'ForegroundColor','g');
            set(handles.str, 'String','Done');
            figure(1);
            plot(Z,AF);
            title('Nonuniformly-Fed linear antenna array | Dolph-Tshebysceff Arrays - array factor vs Z');
            xlabel('Z','fontsize',10);
            ylabel('AF','fontsize',10);
            figure(2);
            polar(Theta1,AF,'-b');
            hold on;
            polar(Theta2,AF,'-b');
            view([90 90]);
            title('Nonuniformly-Fed linear antenna array | Dolph-Tshebysceff Arrays - The 2D pattern of the array');
            Phi_3D   = meshgrid(Phi);
            Theta_3D = meshgrid(Theta1);
            AF_3D    = meshgrid(AF);
            X        = AF_3D.*sin(Theta_3D).*cos(Phi_3D');
            Y        = AF_3D.*sin(Theta_3D).*sin(Phi_3D');
            Z_       = AF_3D.*cos(Theta_3D);
            figure(3);
            surf(X,Y,Z_,'EdgeColor','interp','FaceAlpha',0.1);
            axis vis3d;
            axis equal;
            lighting gouraud;
            title('Nonuniformly-Fed linear antenna array | Dolph-Tshebysceff Arrays - The 3D pattern of the array');
        end
    end
```

Figure 8: Code - Sketch.

1.3. Reset

To reboot program and close all figures.

Figure 9: Reset.

```
    % --- Executes on button press in reset_button.
    function reset_button_Callback(hObject, eventdata, handles)
    set(handles.select,         'Value',1);
    set(handles.part1,          'Visible','on');
    set(handles.part2,          'Visible','off');
    set(handles.part3a,         'Visible','off');
    set(handles.part3b,         'Visible','off');
    set(handles.special_case,   'Visible','on');
    set(handles.s1p,            'Visible','off');
    set(handles.s2p,            'Visible','off');
    set(handles.s3p,            'Visible','off');
    set(handles.str,            'Visible','off');
    set(handles.l_1,       'String','0');
    set(handles.d_2,       'String','0');
    set(handles.N_2,       'String','0');
    set(handles.alpha_2,   'String','0');
    set(handles.d_3a,      'String','0');
    set(handles.N_3a,      'String','0');
    set(handles.alpha_3a,  'String','0');
    set(handles.d_3b,      'String','0');
    set(handles.N_3b,      'String','0');
    set(handles.alpha_3b,  'String','0');
    set(handles.Ro_3b,     'String','0');
    close(figure(1));
    close(figure(2));
    close(figure(3));
```

Figure 10: Code - Reset.

1.4. Special Cases

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

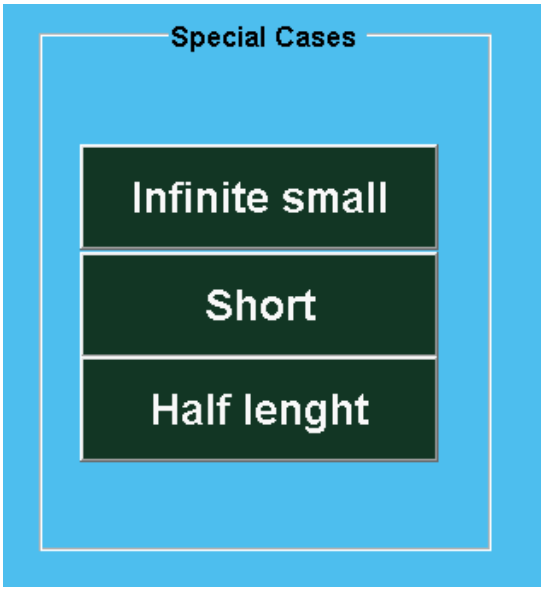


Figure 11: Special Cases.

```
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: $l = 2\lambda$

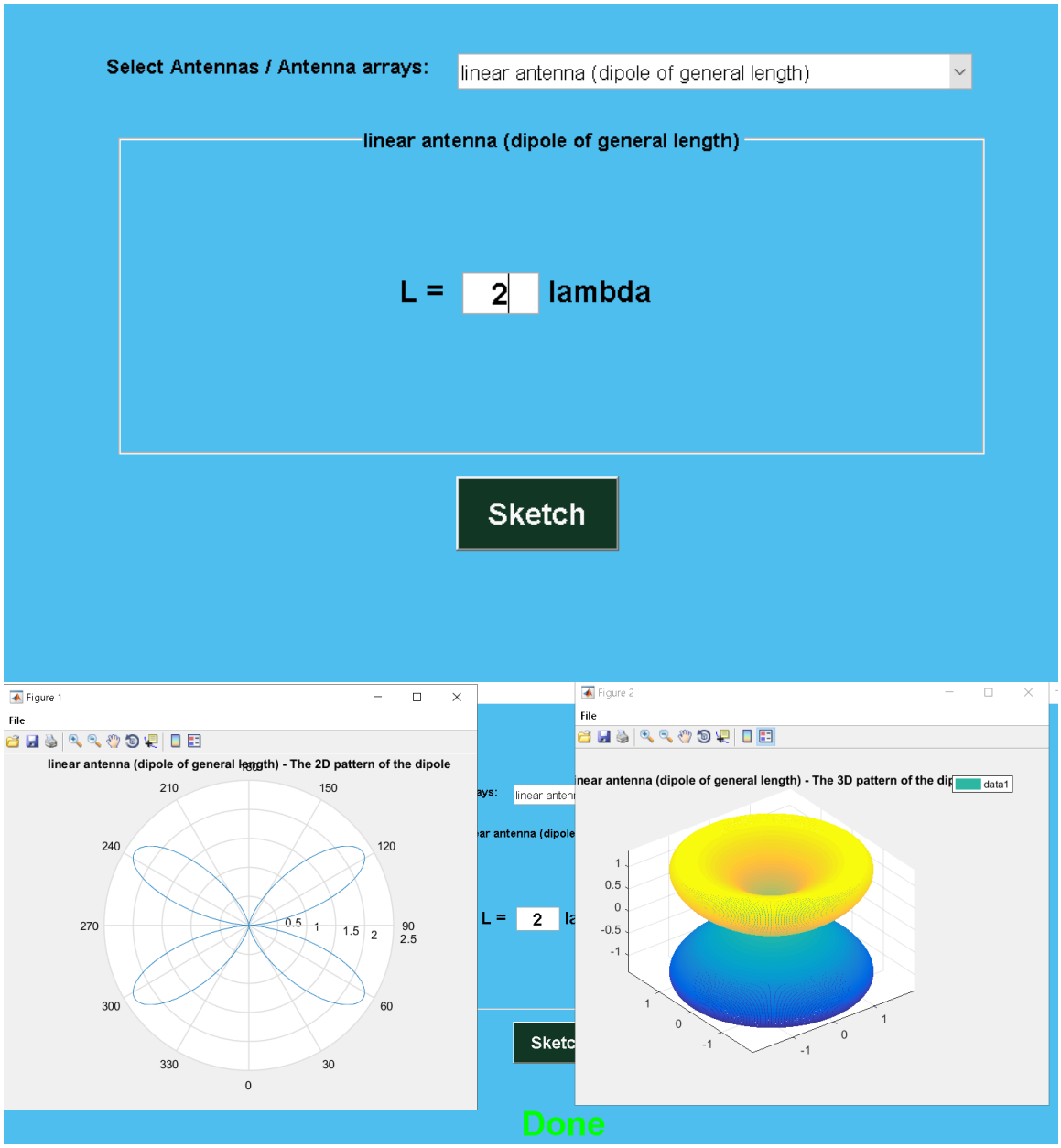


Figure 13: Example 1 - Part 1: linear antenna (dipole of general length).

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

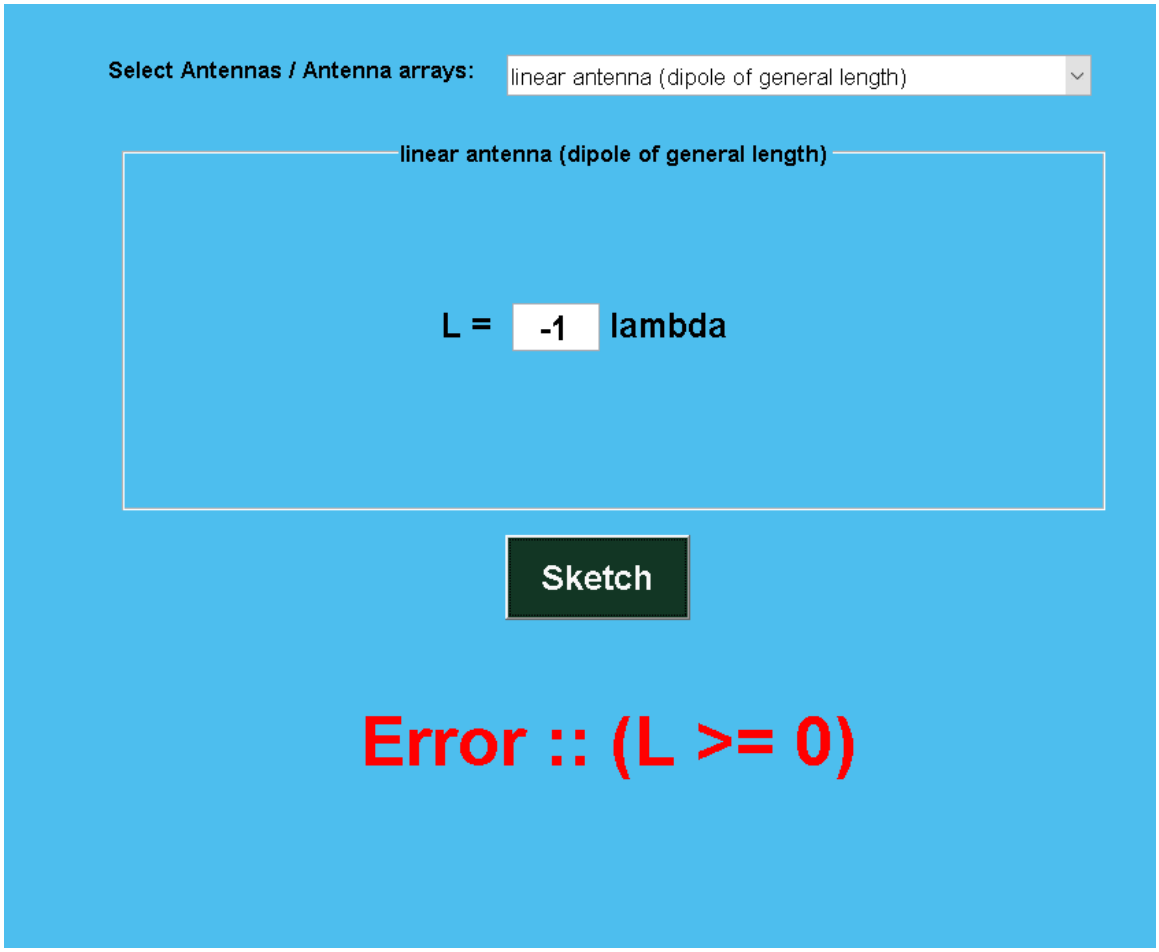


Figure 14: Example 2 (Errors) - Part 1: linear antenna (dipole of general length).

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

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

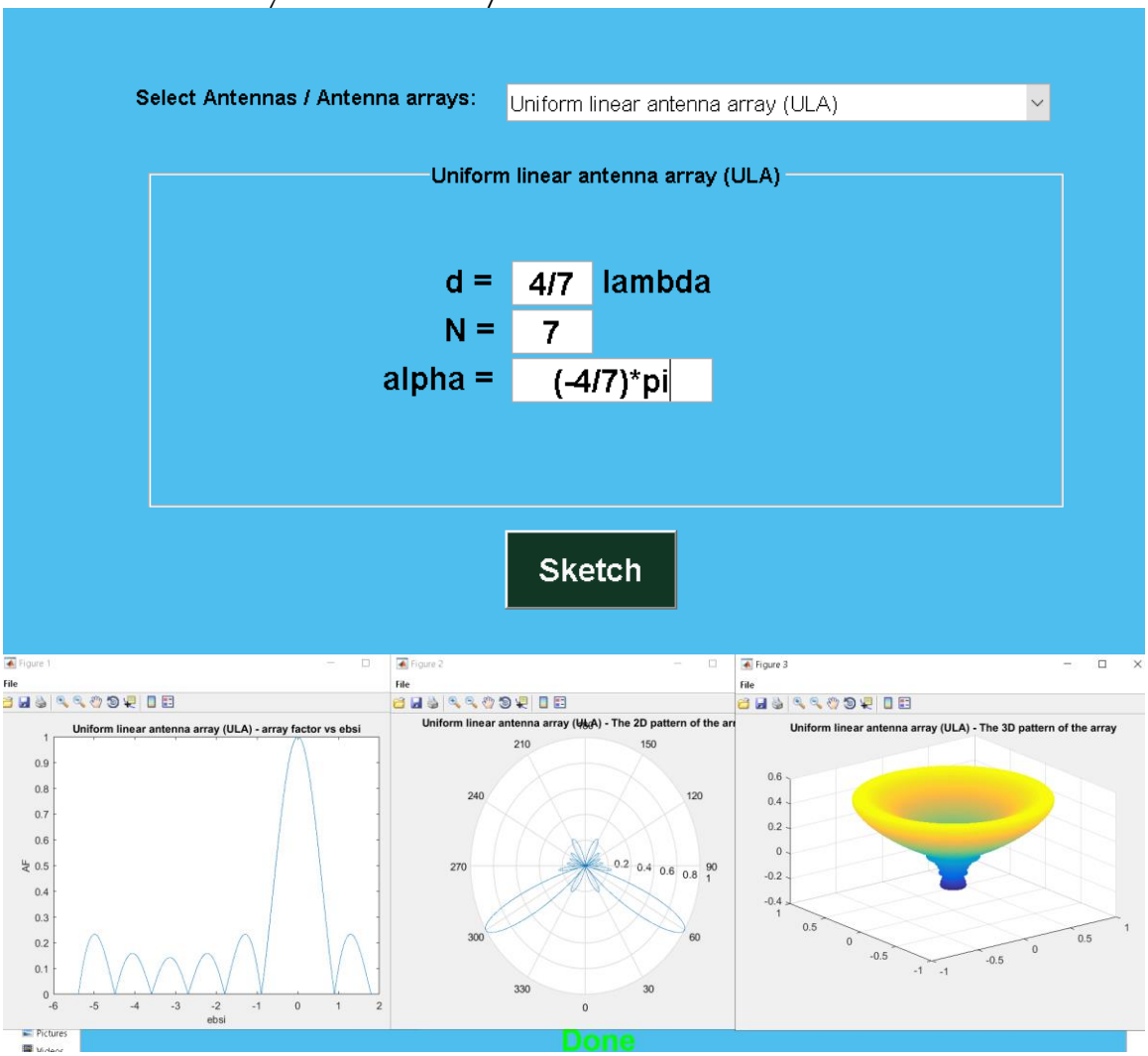


Figure 15: Example 1 - Part 2: Uniform linear antenna array (ULA).

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

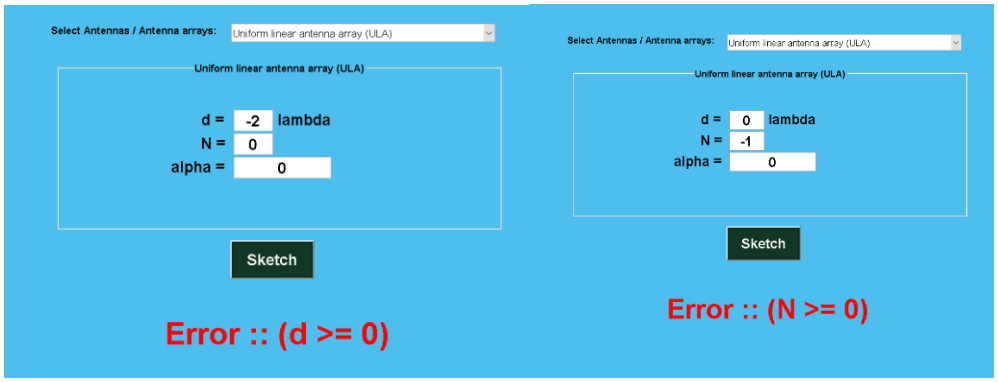
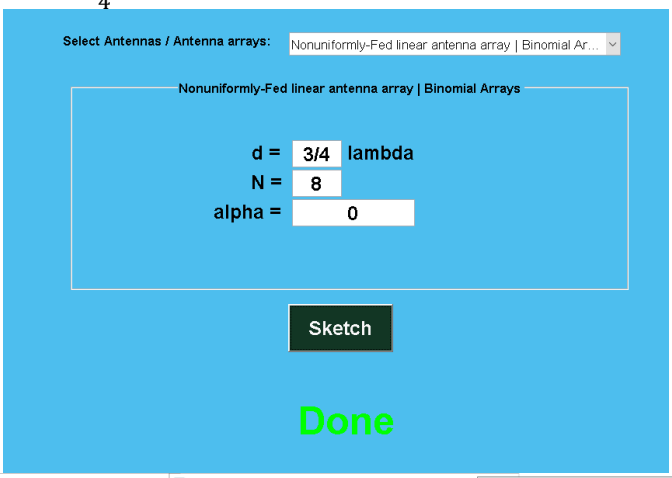


Figure 16: Example 2 (Errors) - Part 2: Uniform linear antenna array (ULA).

2.3. Part 3: Nonuniformly-Fed linear antenna array

A. Binomial Arrays

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



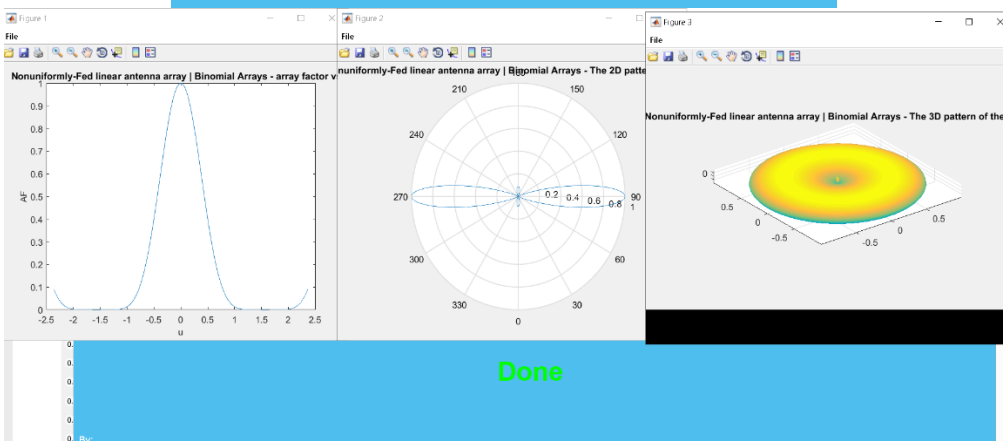


Figure 17: Example 1 - Part 3: Nonuniformly-Fed linear antenna array (Binomial Arrays).

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

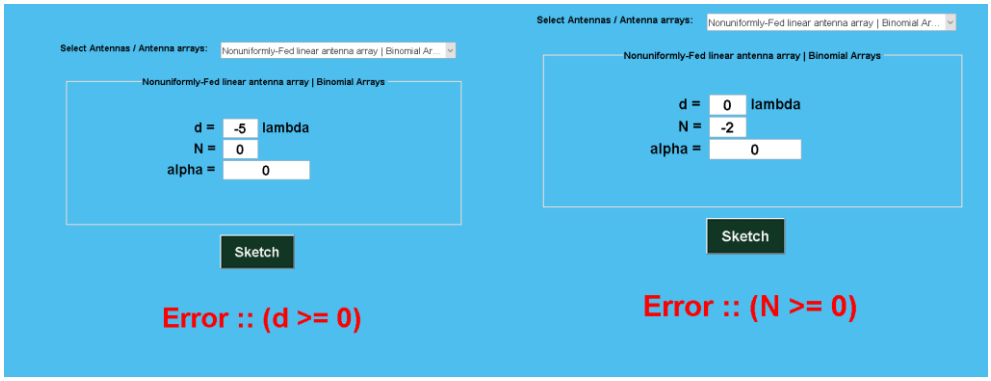
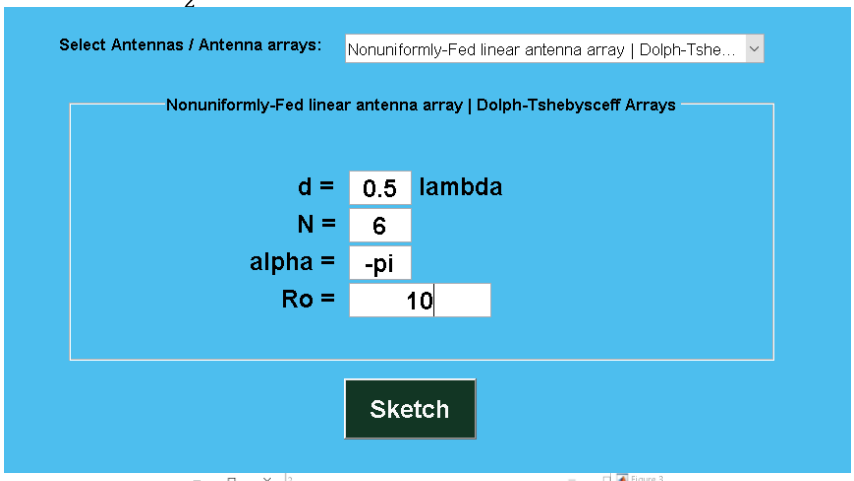


Figure 18: Example 2 (Errors) - Part 3: Nonuniformly-Fed linear antenna array (Binomial Arrays).

B. Dolph-Tschebysceff Arrays

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



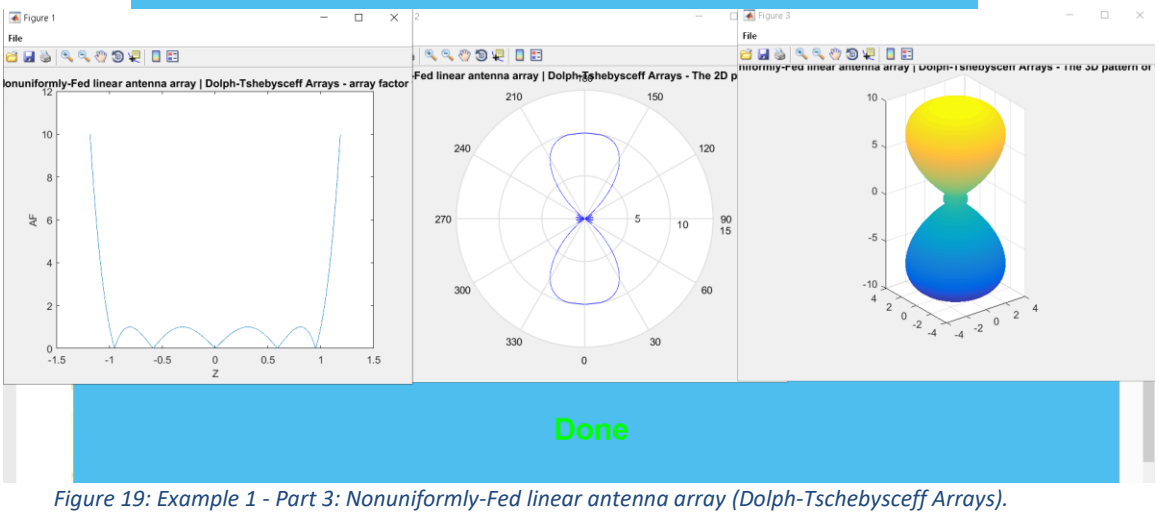


Figure 19: Example 1 - Part 3: Nonuniformly-Fed linear antenna array (Dolph-Tschebysceff Arrays).

Example 3: $Errors$
$if\ d < 0\ or\ N < 0$, we will ask you again.
$Always\ R_0 > 1$.

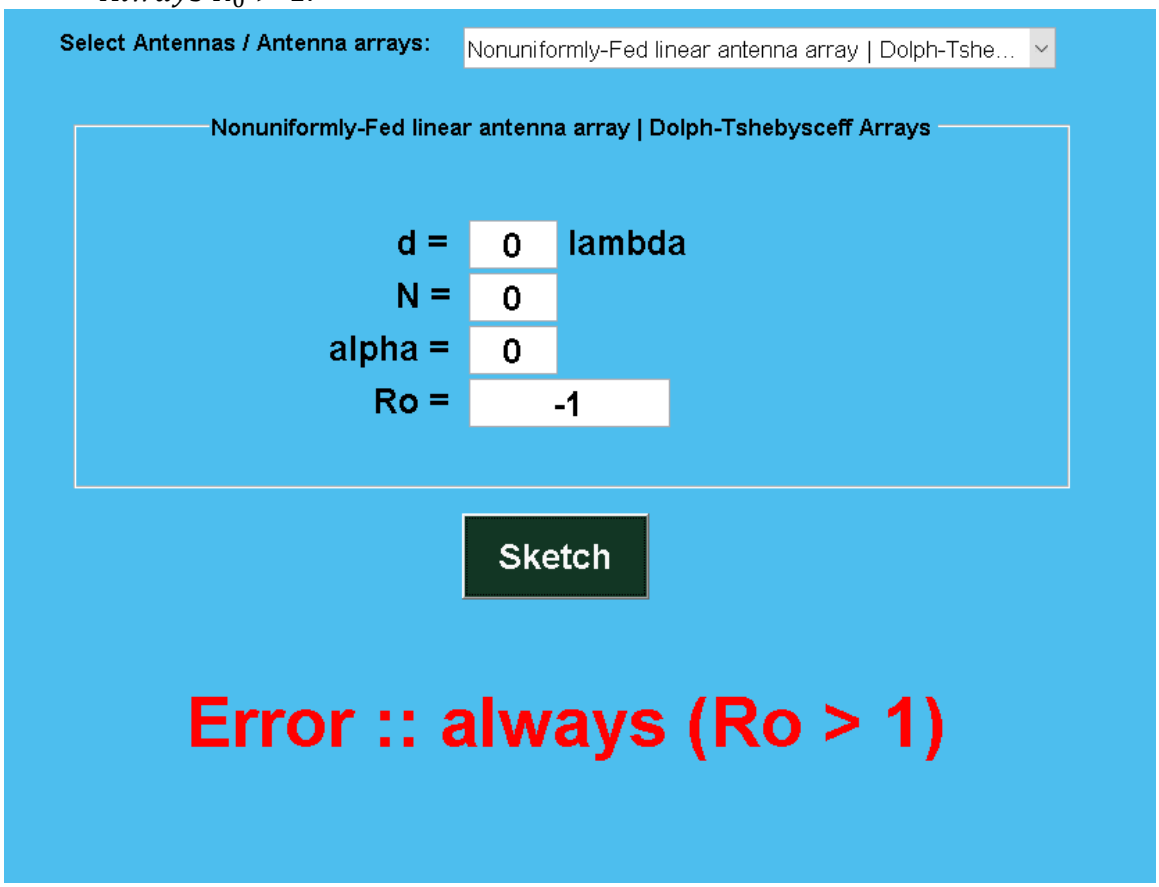


Figure 20: Example 2 (Errors) - Part 3: Nonuniformly-Fed linear antenna array (Dolph-Tschebysceff Arrays).

2.4. Special Case for dipole antenna

A. Infinite small dipole

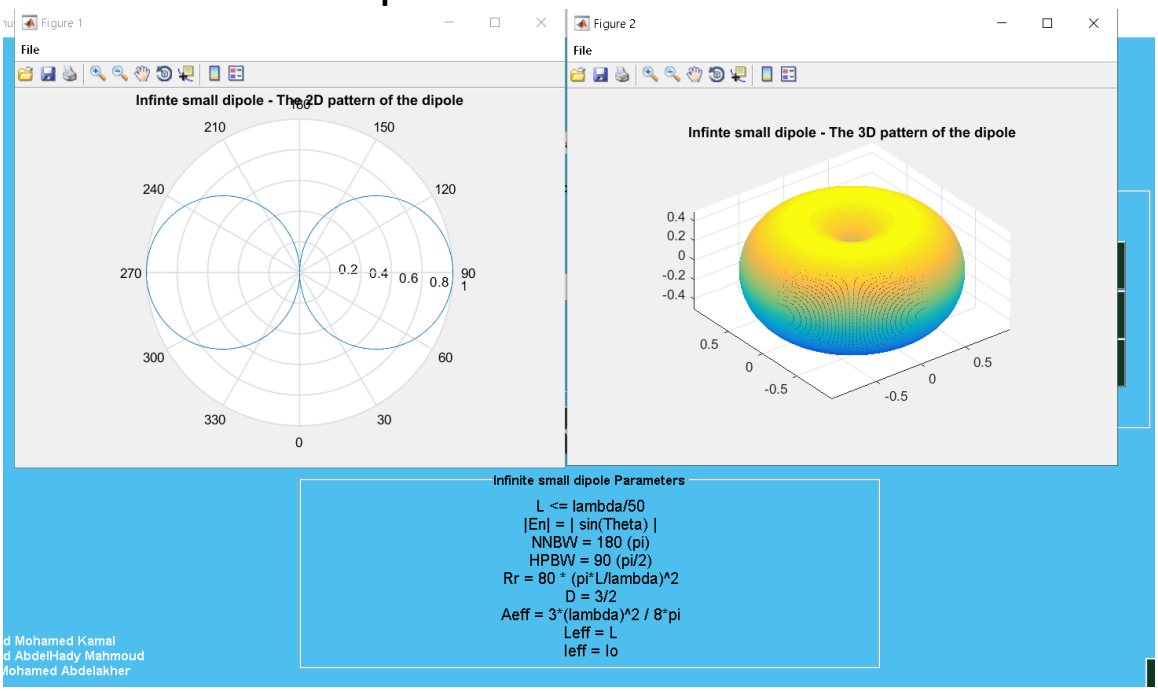


Figure 21: Infinite small dipole.

B. Short dipole

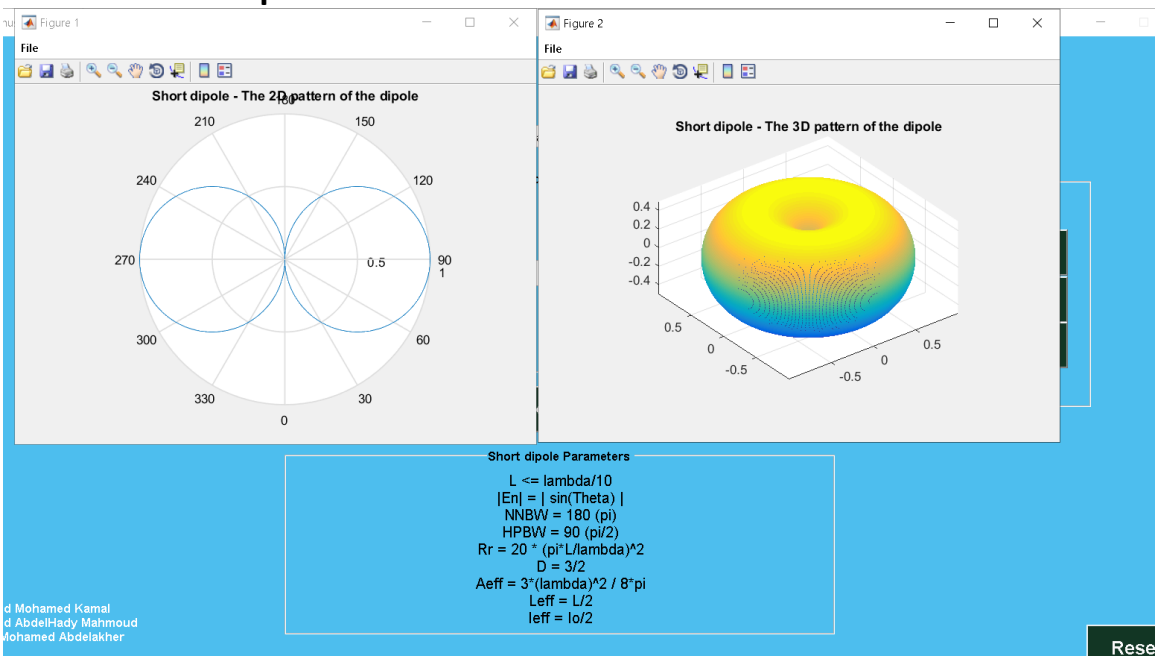


Figure 22: Short dipole.

C. λ/2 dipole

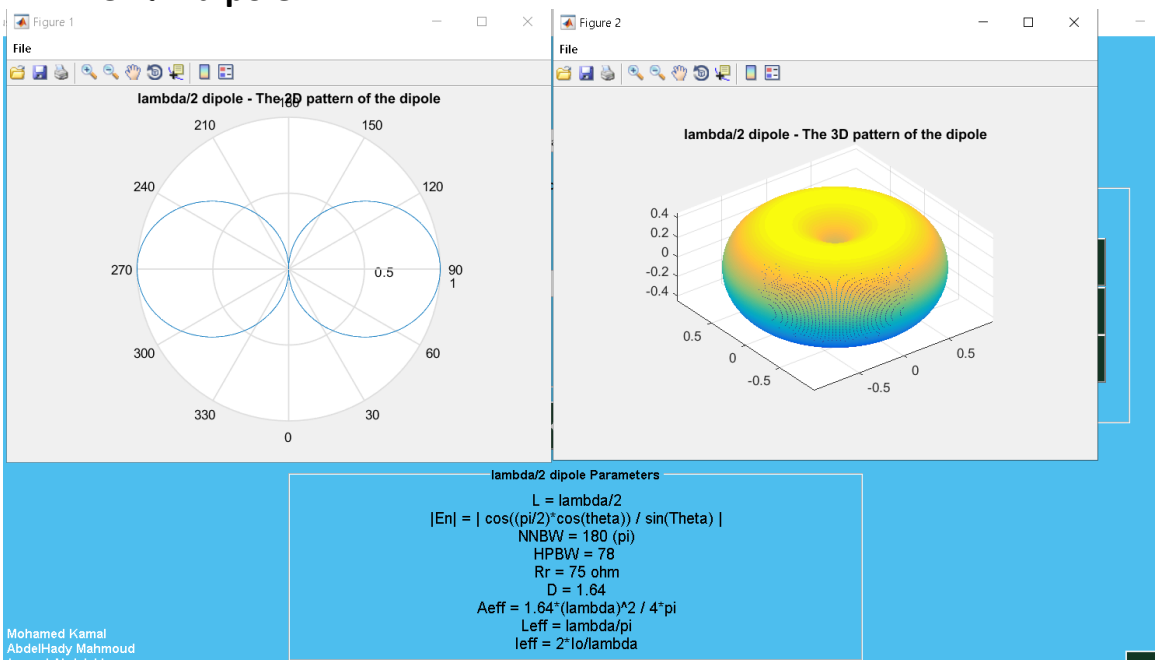


Figure 23: λ/2 dipole.