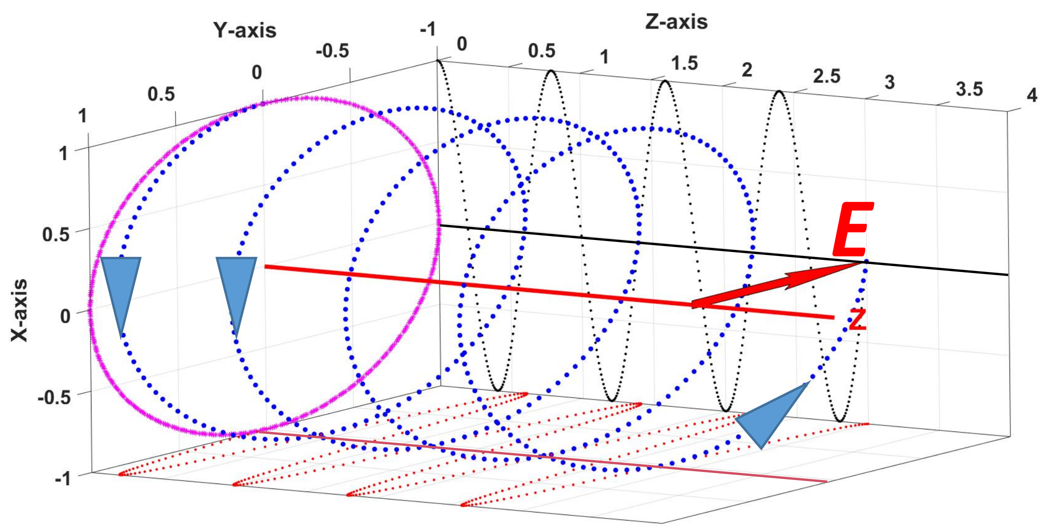
To boost the flavor of electromagnetics, we accompanied many problems with short Matlab code acting as *a calculator and result illustrator*. We completely ignored the description of algorithms behind the codes because they are often quite complicated and required ample, very sophisticated as well as abstract mathematical specifics poorly helping to understand the physical picture. Such approach lets us shift the focus from the often emasculated and practically fruitless problems primarily based on math transformations to more complicated but close to practical tasks. We hope that it helps to visualize typically invisible EM fields and stimulate in-depth analysis and the following discussion of the fundamental principles. Projecting images onto a big screen, the lectures may organize whole class conversation making the audience to be significantly engaged. These scripts are for people who have never programmed in Matlab or CST before.
We cheer our readers to look through Matlab scripts and CST models to discover and sophisticate the algorithm embedded in them. We encourage to use the student edition of MATLAB and CST STUDIO SUITE® to get enhanced problem understanding.

Attention. Regrettably, copy and paste into Matlab Command Window saves appropriate Matlab format just in Chrome Web Browser. You have to restart https://emfieldbook.com/ in https://www.google.com/ if your browser is different.

5.1 **EM Wave Polarization**.



Refresh the material of Section 5.1 in Chapter 5. Copy and paste Matlab script bellow into **Matlab Command Window,** enter the input parameters n =1, a and ψ according to expression (5.2) in Chapter 5, and run it. The full-screen Figure 1 animates the space behavior of E-vector matching three consecutive wavefronts. The blue vector corresponds to the wavefront at $t = 0$, $z = 0$, red one follows behind and reaches the cross section $z = 0$ later with a delay of Δt, while the green one comes to the same cross section $z = 0$ at $t = 2\Delta t$. The matching color thick dots trace the movement of each vector tip along the helical path forwards the z-axis. The magenta, red and black dots in $\theta\varphi-, \theta Z-$ and $\varphi Z-$ plane demonstrate correspondingly the behavior of the total E-vector (polarization ellipse) and its harmonic components $E_\theta(z)$ and $E_\varphi(z)$. The larger diameter dots of the corresponding color depicts the components' magnitude at $z = 0$. Finally, the image in Figure 1 is rotated to be comparable to Figure 5.1.4 in Chapter 5[1]. Please check and understand all the cases mentioned in Section 5.1.1.

[1] Excellent animations can be found at Prof. Ian Copper's website http://www.physics.usyd.edu.au/teach_res/mp/doc/cemPol1.htm, and these two https://www.youtube.com/watch?v=Q0qrU4nprB0 and https://www.youtube.com/watch?v=8YkfEft4p-w.

```
close all, clc; clear; a=input('Enter Magnitude of Current Ratio a = ')
psi=input('Enter Phase (|psi|<= pi/2) of Current Ratio [rad] psi = ')
n=input('Enter coefficient 0.9<= n <=1.1 giving differencies in propagation coefficients = ')
H=6*pi; omega_t=linspace(0,pi,5e0); gamma_z=linspace(0,H,1.5e2); x = ([-1, 1/8, 1/16, 1, 1/16, 1/8,-1]+1)/2;
y = [-1/20,-1/20,-1/10, 0, 1/10, 1/20, 1/20]/3;
for i=1:length(omega_t); for j=1:length(gamma_z); Exx=x*cos(omega_t(i)-gamma_z(j));
    Exy=y*sin(omega_t(i)-n*gamma_z(j)+psi); Eyx=y*sin(omega_t(i)-gamma_z(j));
    Eyy=x*cos(omega_t(i)-n*gamma_z(j)+psi); Ex(i,j,:)=Exx+Eyx; Ey(i,j,:)=a*(Exy+Eyy); end; end;
figure('units','normalized','outerposition',[0 0 1 1]); hold on; box on;
grid on; g1 = hgtransform; g2 = hgtransform; g3 = hgtransform;
text(0,0,1.25*H,'\bfZ','FontSize',45,'Color','k'); xlabel('\bf\Theta-axis'); ylabel('\bf\phi-axis');
ht(1)=text(1,-1,1,'\bfE_\theta vs. Z-coordinate'); set(ht(1),'FontSize',25,'Rotation',90)
ht(2)=text(-1,1,1,'\bfE_\phi vs. Z-coordinate'); set(ht(2),'FontSize',25,'Rotation',90,'Color','r')
v1=[0,0,0]; v2=[0,0,1.1*H]; v=[v2;v1]; hp=plot3(v(:,1),v(:,2),v(:,3),'k'); set(hp,'LineWidth',8);
r=-linspace(0,pi,10); th=linspace(0,2*pi,20); [R,T]=meshgrid(r,th);
X=R.*cos(T)/25; Y=R.*sin(T)/25; Z=R+H*1.25; s=surf(X,Y,Z); set(s,'FaceColor',[0 0 0]);
  for j=1:length(gamma_z); xlim([-1 1]); ylim([-1 1]); zlim([0 1.25]*H); view(144,33); axis square
     hpt1=patch('XData',Ex(1,j,:),'YData',Ey(1,j,:),'FaceColor','b','Parent',g1);
     hpt2=patch('XData',Ex(2,j,:),'YData',Ey(2,j,:),'FaceColor','r','Parent',g2);
     hpt3=patch('XData',Ex(3,j,:),'YData',Ey(3,j,:),'FaceColor','g','Parent',g3);
     g1.Matrix = makehgtform('translate',[0,0,gamma_z(j)]); g2.Matrix = makehgtform('translate',[0,0,gamma_z(j)]);
     g3.Matrix = makehgtform('translate',[0,0,gamma_z(j)]); h1=g1.Matrix; h2=g2.Matrix; h3=g3.Matrix;
     scatter3(hpt1.XData(4),hpt1.YData(4),h1(3,4),140,'b','filled');
     scatter3(hpt2.XData(4),hpt2.YData(4),h2(3,4),140,'r','filled');
     scatter3(hpt3.XData(4),hpt3.YData(4),h3(3,4),140,'g','filled');
     if j==1; M=100; else; M=20; end; scatter3(hpt1.XData(4),hpt1.YData(4),0,10,'m*');
     scatter3(-1,hpt1.YData(4),h1(3,4),M,'r','filled'); scatter3(hpt1.XData(4),-1,h1(3,4),M,'k','filled');
     scatter3(hpt2.XData(4),hpt2.YData(4),0,10,'m*'); scatter3(-1,hpt2.YData(4),h2(3,4),M,'r','filled');
     scatter3(hpt2.XData(4),-1,h2(3,4),M,'k','filled'); scatter3(hpt3.XData(4),hpt3.YData(4),0,10,'m*');
     scatter3(-1,hpt3.YData(4),h3(3,4),M,'r','filled'); scatter3(hpt3.XData(4),-1,h3(3,4),M,'k','filled');
     drawnow; if j~=length(gamma_z); delete(hpt1); delete(hpt2); delete(hpt3); end; end; delete(ht); camroll(-130);
```

1. What does the term polarization mean? How is this term connected to E- and H-vector orientation? May the polarization define as the direction of EM wave propagation?

2. Come back to Section 3.1.8 in Chapter 3 and explain the polarization effect through the photons migration.

3. What could you tell looking at Figure 1 about the motion of E-vector tip?

4. What happens with polarization when two collocated linear polarized orthogonal electric dipoles emit EM waves? How to keep the total polarization linear? What does it mean slant polarization?

5. Can the polarization be rectangular? If the answer is yes, explain how to create it. If the answer is no, then why (*Hint.* Recall that E- and H-fields are related to each other by differential operations).

6. What are the differences between RHCP and LHCP? Run Matlab script to check your answer.

7. Prove analytically using the expression (5.4) and (5.6) that the superposition of LHCP and LHCP waves propagating in the same direction with equal speed can form the joint EM wave of linear or elliptically polarization.

8. What kind of polarization could form two linear polarized waves propagating in the same direction with equal speed and magnitude? Run Matlab script to check your answer.

9. What is the polarization of the propagating wave if $E_\theta = E_0\cos(\omega t - kz)$ and $E_\varphi = \pm E_0\sin(\omega t - kz)$? Run Matlab script to check your answer.

10. The same question as #9 as $E_\theta = E_0\cos(\omega t - kz)$ and $E_\varphi = \pm 2E_0\sin(\omega t - kz)$? Run Matlab script to check your answer. Do the projections in θZ- , φZ- , and $\theta\varphi$-plane reflect correctly the harmonic nature of the propagating wave?

11. The same question as #9 as $E_\theta = E_0\cos(\omega t - kz)$ and $E_\varphi = \pm 2E_0\cos(\omega t - kz)$? Run Matlab script to check your answer. Do the projections in θZ- , φZ- , and $\theta\varphi$-plane reflect correctly the harmonic nature of the propagating wave?

12. The same question as #9 as $E_\theta = E_0\cos(\omega t - kz)$ and $E_\varphi = E_0\cos(\omega + kz)$? Run Matlab script to check your answer. Do the projections in θZ- , φZ- , and $\theta\varphi$-plane reflect correctly the harmonic nature of the propagating wave?

13. Why do we need to match receiving and transmit antennas polarization? What kind of polarization is better to use for communication if the LP receiving antenna orientation is unknown or might vary?

14. Explain and provide examples of systems where LP is preferable (check page 204 in the book). Does the cost and complexity of the antenna influence the choice of polarization?

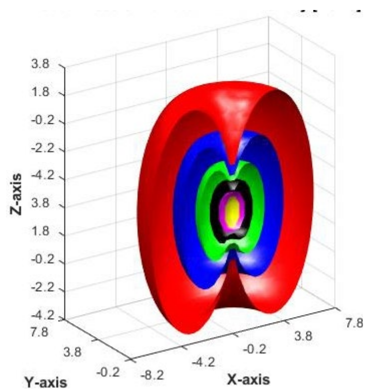
15. How and why can the CP polarization improve the performance of different types of radar and communication systems? Give an example like weather radar, satellite, and cellular communications, etc.

16. Give an example of systems that benefit from applications of the wave with twisted polarization.

17. Explain the terms co-, cross-polarization, and depolarization. What does it mean the axial ratio $AR = 0.5$? Might all these terms be applied to the waves with twisted polarization? Explain.

18. In Section 2.7.4 of Chapter 2 we learned that the propagation coefficient k of EM waves may depend on their structure (check later Section 6.8 in Chapter 6) and particularly of wave polarization (so-called Faraday Effect in ferrite, ionosphere, magneto-optic crystals, etc.). Matlab script allows simulating this phenomenon. Rerun the script entering, for example, $n = 0.9$, $a = 1$, and $\psi = \pi/2$ and watch the CP polarization variation as the wave propagates over the z–axis. Try to explain the results and think how to use them practically for polarization transform. Do the projections in θZ- , φZ- , and $\theta\varphi$-plane continue to be the harmonic waves?

5.2 **Near-field Zone vs. Far-field Zone**. Refresh the material of Section 3.1.5 in Chapter 3, Section 4.3.1 in Chapter 4, Sections 5.2.5 and 5.2.9 in Chapter 5. Note the expression for E_r in (5.20) is not correctly printed and should be rewritten as



$$E_r = E_{r0}\left(\frac{1}{(kr)^2} - \frac{j}{(kr)^3}\right)\cos\theta e^{j(\omega t - kr)}$$

The field magnitudes in (5.20) are equal to $H_{\varphi 0} = 0.5 I_e^{exc} k\,\Delta l/\lambda$, $E_{\theta 0} = 120\pi H_{\varphi 0}$, $E_{\varphi 0} = 240\pi H_{\varphi 0}$ and can be found by applying (5.40) to (4.60). It was assumed that $I_e^{exc} = 1[A]$, $k = 2\pi$. Elementary electric radiator of $\Delta l/\lambda = 0.1$ is oriented along z-axis and enclosed in cube of 80λ x 80λ x 80λ divided uniformly into 211 x 211 x 211 cells of $\cong 0.4\lambda$ x 0.4λ x 0.4λ each. You might change any or all of these parameters editing Matlab script but be careful. The near-fields head very fast to infinity as $r \to 0$ that makes it

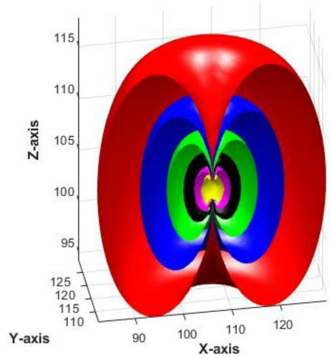


Figure 1

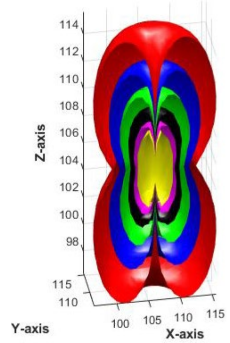


Figure 2

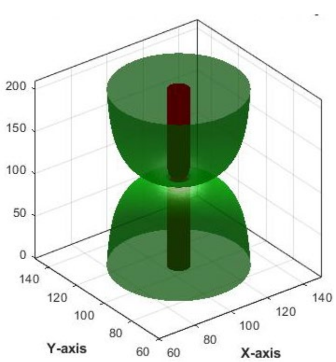


Figure 3

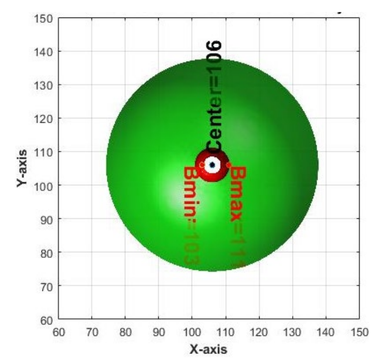


Figure 4

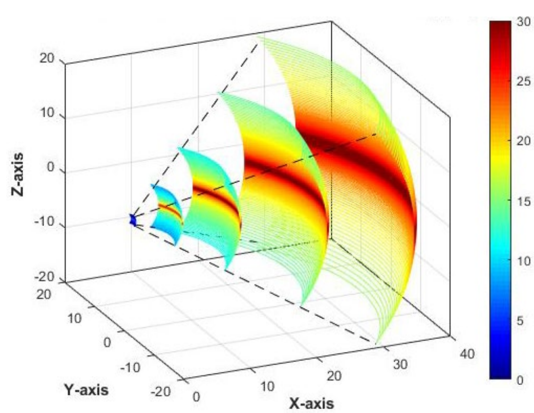


Figure 5

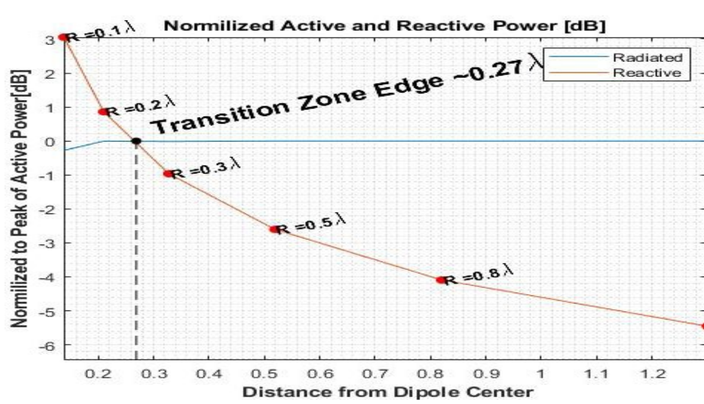


Figure 6

difficult to adjust properly the drawings' appearance, colormap and scaling. Copy and paste Matlab script bellow into **Matlab Command Window.** Six plots like shown above should appear consequently. Figure 1 (top left on the display) demonstrates the *cut in half* isosurfaces of the *radiated power density* normalized to the peak of total power density accumulated inside the cube. Over the entire isosurface, the magnitude of power density is constant and was chosen from 0.0001 (red surface relatively far from radiator) to 0.01 (yellow surface) in 6 points of the logarithmic scale. Figure 2 (bottom left) is the same but for the normalized *reactive/accumulated power density* inside the cube. Figure 3 (top center) depicts the animated isosurfaces showing the ratio of radiated power density to reactive one inside the cube. The ratio magnitude is in logarithmic scale and includes 25 points from 0.1 (radiated = 0.1* reactive) to $10^{0.8}$ ~ 6.3 (radiated = 6.3*reactive) to get smooth animation. The starting animation is colored in red meaning that reactive part exceeds the radiated one. The color switches to green as soon as this ratio surpasses one. The final image includes the red surface symbolizing that the radiated and reactive power densities are equal. The circles in Figure 4 (bottom center) is the top view of the hourglass in Figure 3. Figure 5 (top right) illustrates how the ratio between radiated and reactive power density changes as the EM energy propagates along the *x*-axis from the radiator virtual center at $x = 0$, $y= 0$, $z = 0$. The contour lines on each surface and colorbar nearby give the graduated in dB reading. Keep in mind that the scale on the axes in Figures 1 – 5 is specified into the cell index numbers from the left bottom corner of the cube. To convert any of these indexes into real-world cell position normalized to wavelength follow the expression N*80/211 where 80 is the cube side and 211 is the quantity of each side split. For example, the dipole center is situated at the point marked as **Center** (106,106,106) or ~ $(40.2\lambda, 40.2\lambda, 40.2\lambda)/(2\pi)$ as it is pictured in Figure 4. The red points marked as Bmin and Bmax in Figure 4 is located on the bounding surface where the fields emitted by the dipole transfer smoothly into the near- and far-field zone. Therefore, the transition zone for elementary electric dipole extends between 3 and 5 cell lengths or from $3*80/211/(2\pi) = 0.18\lambda$ to $5*80/211/(2\pi) = 0.3\lambda$.[2] This slightly adjusted value is depicted in Figure 6 (bottom right) that describes the behavior of *radiated* power (blue line), i.e. travelling away and never returning, and *reactive* (red line), i.e. accumulated nearby, along the *average* distance $R = \sqrt{ab}$ from the dipole center where *a* and *b* are minor and major axis of ellipses in Figure 1.

1.

[2] Note that according to Figure 1 and 2 this limit depends on the elevation angle and exceeds that is given typically in the literature for the elementary electrical dipole (see, for example, problem 2.3-5 in W. L. Stutzman, G. A. Thiele, Antenna Theory and Design, 3rd edition, 2013) and our book on page 212.

```
close all; clc; clear; warning('off','all'); I=1; L_Lam=0.1; Lam=1; k=2*pi/Lam; BoxSize=40/Lam; N=2e2+11;
x=linspace(-1,1,N)*BoxSize; y=linspace(-1,1,N)*BoxSize; z=linspace(-1,1,N)*BoxSize;
[X,Y,Z]=meshgrid(x,y,z); [Az,El,R] = cart2sph(X,Y,Z);
Etheta=(60*pi*I*L_Lam*k)*((1i./R+1./R.^2-1i./R.^3).*cos(El).*exp(-1i*R));
Er=(120*pi*I*L_Lam*k)*((1./R.^2-1i./R.^3).*sin(El).*exp(-1i*R)); Ephi=0*Er;
E=sqrt(Etheta.*conj(Etheta)+Er.*conj(Er)); Emax=max(E(:)); E1=20*log10(E/Emax);
Hphi=(I*k*L_Lam/2)*((1i./R+1./R.^2).*cos(El).*exp(-1i*R));
H=sqrt(Hphi.*conj(Hphi)); Hmax=max(H(:)); H1=20*log10(H/Hmax);
Sr=Etheta.*conj(Hphi)/2; Stheta=-Er.*conj(Hphi)/2; SrR=real(Sr); SthetaR=real(Stheta); SR=sqrt(SrR.^2+SthetaR.^2);
SrI=imag(Sr); SthetaI=imag(Stheta); SI=sqrt(SrI.^2+SthetaI.^2); SS=sqrt(SI.^2+SR.^2); SSMax=max(SS(:));
Ratio=SR./SI; RatR=10*log10(real(Ratio)); DR = abs(SR/SSMax); DR(1:N/2,:,:)=NaN;
DI = abs(SI/SSMax); DI(1:N/2,:,:)=NaN;
f1=figure(1); movegui(f1,'northwest'); grid on; jc=0; cc={'r','b','g','k','m','y'}; hold all;
for id=logspace(-4, -2, 6)
   jc=jc+1; isoR(jc)=id; [fc,vr]=isosurface(DR,id); a=vr(fc(:,2),:)-vr(fc(:,1),:); b=vr(fc(:,3),:)-vr(fc(:,1),:);
   c=cross(a,b,2); area=sum(sqrt(sum(c.^2, 2)));
   pR=patch('Faces',fc,'Vertices',vr,'FaceColor',char(cc(jc)),'EdgeColor','none'); PowerR(jc)=id*area;
   a1=max(pR.Vertices(:))-106; b1=abs(min(pR.Vertices(:))-106); Rho(jc)=sqrt(a1*b1*80/211/pi)/2/pi; end
title('\bfIsosurfaces of Radiated Power Density [W/m^2]');
xlabel('\bfX-axis'); ylabel('\bfY-axis'); zlabel('\bfZ-axis'); axis tight;
view(-10,33); daspect([1,1,.4]); camlight left; camlight; lighting gouraud;
f2=figure(2); movegui(f2,'southwest'); grid on; jc=0;
for id=logspace(-4, -2, 6); jc=jc+1; isoI(jc)=id; [fc1,vr1]=isosurface(DI,id);
   pI = patch('Faces',fc1,'Vertices',vr1,'FaceColor',char(cc(jc)),'EdgeColor','none');
   a = vr1(fc1(:,2),:)-vr1(fc1(:,1),:); b = vr1(fc1(:,3),:)-vr1(fc1(:,1),:); c = cross(a,b,2); area=sum(sqrt(sum(c.^2,2)));
   PowerI(jc)=id*area; end
xlabel('\bfX-axis'); ylabel('\bfY-axis'); zlabel('\bfZ-axis');
title('\bfIsosurfaces of Reactive Power Density [W/m^2]'); axis tight;
view(-10,33); daspect([1,1,.4]); camlight left; camlight; lighting gouraud
f3=figure(3); box on; hold all; grid on; vals = logspace(-1, 0.8, 25);
[faces,verts,colors] = isosurface(Ratio,vals(1),R);
h=patch('Vertices',verts,'Faces',faces,'FaceVertexCData',colors,'FaceColor','interp','EdgeColor','none'); alpha(0.7);
axis([60  150 60 150 0 210]); xlabel('\bfX-axis'); ylabel('\bfY-axis'); camlight; lighting phong;
for id = vals; [fc2,vr2]=isosurface(Ratio,id); if id<1
   set(h, 'Faces', fc2, 'Vertices', vr2,'facecolor', 'r', 'edgecolor', 'none'); else
   set(h, 'Faces', fc2, 'Vertices', vr2,'facecolor', 'g', 'edgecolor', 'none'); end
  pause(0.2); view(3); axis square; end
xlabel('\bfX-axis'); ylabel('\bfY-axis'); title('\bfRatio of Radiated to Reactive Power Density')
[fc3,vr3]=isosurface(Ratio,1); pR1= patch('Faces',fc3,'Vertices',vr3,'FaceColor','r','EdgeColor','none');
Ax=get(gca,'Xtick'); Ay=get(gca,'Ytick'); Az=get(gca,'Ztick'); savefig(f3,'Poyn.fig');
ho=openfig('Poyn.fig'); movegui(figure(4),'south'); view(0,90); hsc = scatter3(106,106,117,'MarkerFaceColor','k');
hsc.SizeData = 20; hsc = scatter3(111,106,117,'MarkerFaceColor','r'); hsc.SizeData = 20;
hsc = scatter3(103,106,117,'MarkerFaceColor','r'); hsc.SizeData = 20;
text(114,106,117,'Bmax=111','Color','r','FontSize',18,'FontWeight','Bold','Rotation',-90);
text(106,109,117,'Center=106','Color','k','FontSize',18,'FontWeight','Bold','Rotation',90);
text(100,106,117,'Bmin=103','Color','r','FontSize',18,'FontWeight','Bold','Rotation',-90);
f5=figure(5); movegui(f5,'northeast'); grid on; box on; hold all; colormap(jet(80)); view(-24,26); colorbar; j1=0;
for jj=[.02 .1 0.2 0.35 0.5] ; j1=j1+1; [ysurf,zsurf] = meshgrid(y*jj,z*jj);
   rho=sqrt(max(abs(ysurf(:)))^2+max(abs(zsurf(:)))^2)*sqrt(2); xsurf=sqrt(abs(rho^2-ysurf.^2-zsurf.^2));
   hcs=contourslice(X,Y,Z,RatR,xsurf,ysurf,zsurf,150); cor1(j1,:)=hcs(1).Vertices(1,:); cor2(j1,:)=hcs(2).Vertices(1,:);
   cor3(j1,:)=hcs(3).Vertices(1,:); cor4(j1,:)=hcs(4).Vertices(1,:);     caxis([0 30]); drawnow; pause(0.2); end
title('\bfNormilized Ratio of Radiated to Reactive Power Density [dB]')
xlabel('\bfX-axis'); ylabel('\bfY-axis'); zlabel('\bfZ-axis');
plot3([cor1(1,1) cor1(j1,1)],[cor1(1,2) cor1(j1,2)],[cor1(1,3) cor1(j1,3)],'k--')
plot3([cor2(1,1) cor2(j1,1)],[cor2(1,2) cor2(j1,2)],[cor2(1,3) cor2(j1,3)],'k--')
plot3([cor3(1,1) cor3(j1,1)],-[cor3(1,2) cor3(j1,2)],[cor3(1,3) cor3(j1,3)],'k--')
plot3([cor4(1,1) cor4(j1,1)],-[cor4(1,2) cor4(j1,2)],[cor4(1,3) cor4(j1,3)],'k--'); shading interp;
f6=figure(6); movegui(f6,'southeast');
PRe=10*log10(PowerR/max(PowerR)); PIm=10*log10(PowerI/max(PowerR));
px=(Rho(4)+Rho(5))/2; plot(Rho,PRe,Rho,PIm); hold all;
plot(px+0*Rho,linspace(min(PIm)-1,0,length(Rho)),'--k'); grid minor; ylim([min(PIm)-1 4]);
hsc = scatter(px,0,'filled','k'); hsc.SizeData = 30;
text(px*1.1,0.4,'Transition Zone Edge ~0.27\lambda','Color','k','FontSize',15,'FontWeight','Bold','Rotation',15);
for jj=1:6; scatter(Rho(jj),PIm(jj),'filled','r'); hsc.SizeData = 30;
   ht=text(Rho(jj),PIm(jj),['R =',num2str(round(Rho(jj),1)),'\lambda']);
   set(ht,'Color','k','FontSize',10,'FontWeight','Bold','Rotation',15); end
title('\bfNormilized Active and Reactive Power [dB]'); axis tight; ylabel('\bfNormilized to Peak of Active Power[dB]');
xlabel('\bf Distance from Dipole Center'); legend('Radiated','Reactive')
```