Приложение 1.1

Расчет значений поляриметрических переменных в MATLAB:
ледяные кристаллы столбчатого типа, распределенные в пространстве хаотически

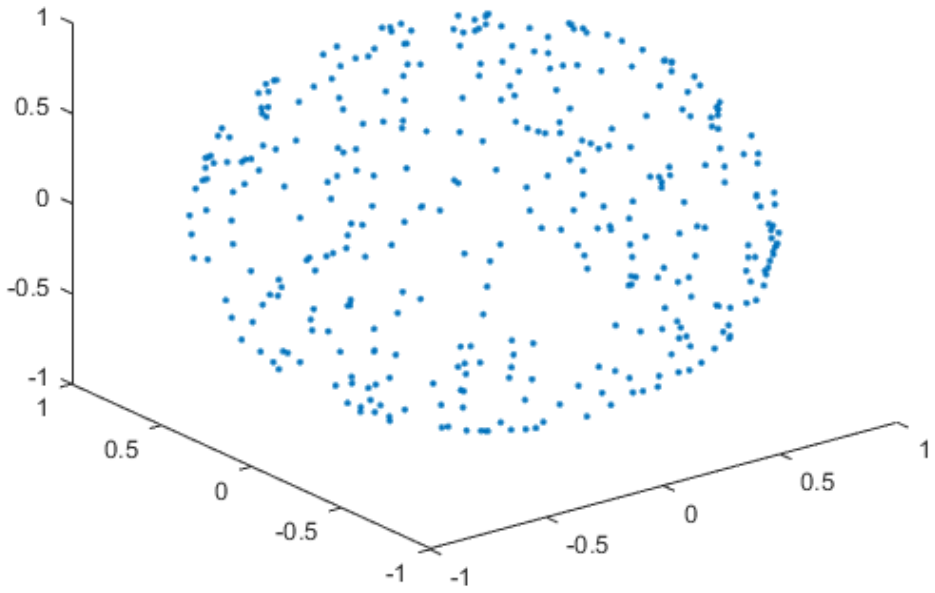
Оглавление

Устанавливаем начальные значения основных параметров

```
clear;
clc;
global wavelength;
global eps_r;
wavelength=32; % задаем длину волны в мм
eps_r=3; % задаем значение диэлектричекой проницаемости

ZDR_crystals(10,1)=zeros; % массив нулей для последующего расчета ДО
LDR_crystals(10,1)=zeros; % массив нулей для последующего расчета ЛДО
Rho_crystals(10,1)=zeros; % массив нулей для последующего расчета КК

n=200000; % общее число точек
k_min=n; % минимальное количество точек
s=0; % переменная-переключатель, используется для отрисовки распределения
    % точек на поверхности единичной сферы только 1 раз

 Phh_crystals(10,n)=zeros; % массив нулей для последующего расчета
                    % коэффициентов формы
 Phv_crystals(10,n)=zeros;
 Pvv_crystals(10,n)=zeros;
 j=1; % эта переменная используется для индексации значений массивов
     % при хранении расчетов для разных углов места

for teta_deg=0:20:180 % перебираем углы места наклона антенны радиолокатора)
```

Задаем распределение точек на поверхности единичной сферы

```
for L=0.1:0.1:8 % задаем размер кристалла

 r=1; % радиус шара с центром в (0,0,0)
 X=zeros(1,n);% массивы хранения координат точек
 Y=zeros(1,n);
 Z=zeros(1,n);
 k=0; % счетчик точек, попавших внутрь шара
 Er=0.01; % погрешность
  for i=1:n
       x=randn/pi; % новая возможная координата по оси X
       y=randn/pi; % новая возможная координата по оси Y
       z=randn/pi; % новая возможная координата по оси Y
       % таким образом задается равновероятное распределение

       % z=(1-2*rand);
       ro2=x^2+y^2+z^2; % квадрат расстояния от точки до центра
       if abs(ro2-r^2)<=Er % ограничение по радиусу
           k=k+1;% данная точка попала в круг
           X(k)=x;
           Y(k)=y;
           Z(k)=z;
       end
   end
  if k<k_min
      k_min=k;
  end
 X=X(1:k_min); % удаляем ненужные нули
 Y=Y(1:k_min);
 Z=Z(1:k_min);
  if s==0 % проверка, был ли данный график уже нарисован
    plot3(X,Y,Z,'.')
    % в результате получим изображение распределения точек по
    % поверхности единичной сферы. Здесь видно, что оно - равномерное
  end
```



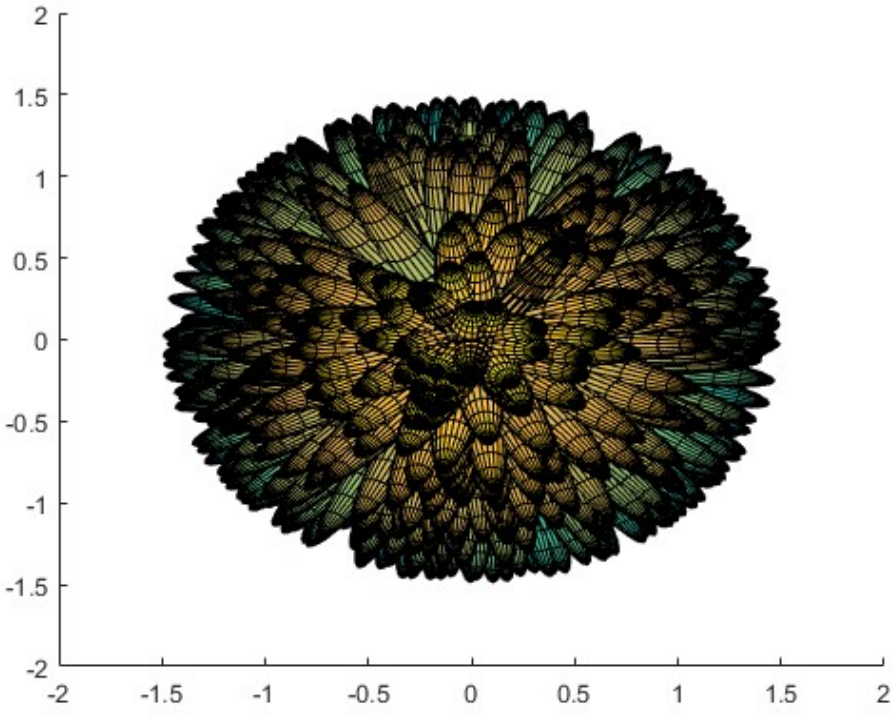
Преобразовываем систему координат

```
[alphaU deltaU R]=cart2sph(X,Y,Z); % переходим от Декартовой системы
                                   % координат к сферической


alpha=pi/2+alphaU;
delta=3*pi/2-deltaU;
% преобразовываем углы alpha и delta в соответствии с теми,
% что используются в формулах 2.51-2.53 главы 2 диссертации
phi=0; % задаем угол поворота горизонтальной плоскости поляризации
teta=teta_deg*pi/180; % переводим из градусов в радианы
```

Рисуем совокупность кристаллов с заданным распределением в пространстве

```
if s==0  % проверка, был ли данный график уже нарисован
    figure;
    hold on
    for i=1:length(alpha)
      illustrate(alpha(i),delta(i),0,0,3);
      % эта функция рисует эллипсоид с заданными соотношением сторон
      % и положением в пространстве
    end
    axis([-2 2 -2 2 -2 2]);
end
```

```
   s=1;
   % На этом графике также видно, что распределение - равномерное
```



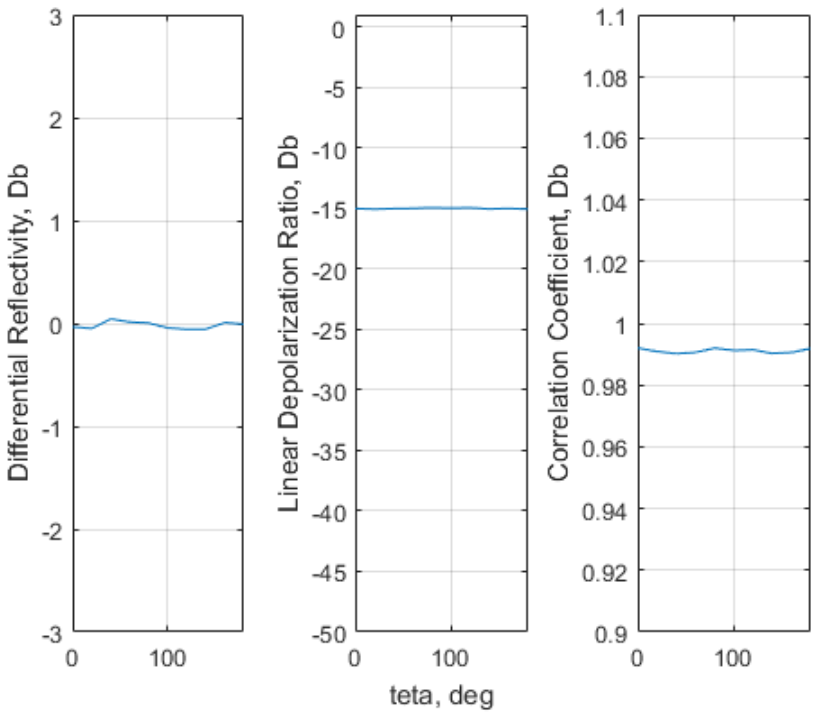
Вычисляем значения поляриметрических параметров

```
   for i=1:k_min
         [hh vv hv]=q(alpha(i),delta(i),phi,teta,L);
           Phh_crystals(j,i)=Phh_crystals(j,i)+hh*N_fun(L)*fun_sigma(L);
           Pvv_crystals(j,i)=Pvv_crystals(j,i)+vv*N_fun(L)*fun_sigma(L);
           Phv_crystals(j,i)=Phv_crystals(j,i)+hv*N_fun(L)*fun_sigma(L);
% Phh_crystals(j,i)=Phh_crystals(j,i)+hh*N_fun(fun_d(L))*fun_sigma(fun_d(L));
% Pvv_crystals(j,i)=Pvv_crystals(j,i)+vv*N_fun(fun_d(L))*fun_sigma(fun_d(L));
% Phv_crystals(j,i)=Phv_crystals(j,i)+hv*N_fun(fun_d(L))*fun_sigma(fun_d(L));
%             Закоментированные три строки выше используются для
%             моделирования отражения от пластинчатых кристаллов
      end

 end
   ZDR_crystals(j,1)=10*log10(sum(Phh_crystals(j,1:k_min))/sum(Pvv_crystals(j,1:k_min)));
   % Вычисляем значения ДО для совокупности частиц
LDR_crystals(j,1)=10*log10(sum(Phv_crystals(j,1:k_min))/sum(Pvv_crystals(j,1:k_min)));
   % Вычисляем значения ЛДО для совокупности частиц
Rho=corrcoef(Phh_crystals(j,:),Pvv_crystals(j,:));
Rho_crystals(j)=Rho(1,2);
   % Вычисляем значения КК для совокупности частиц
 j=j+1;
```

```
end
% Переходим к расчетам для следующего угла места
```

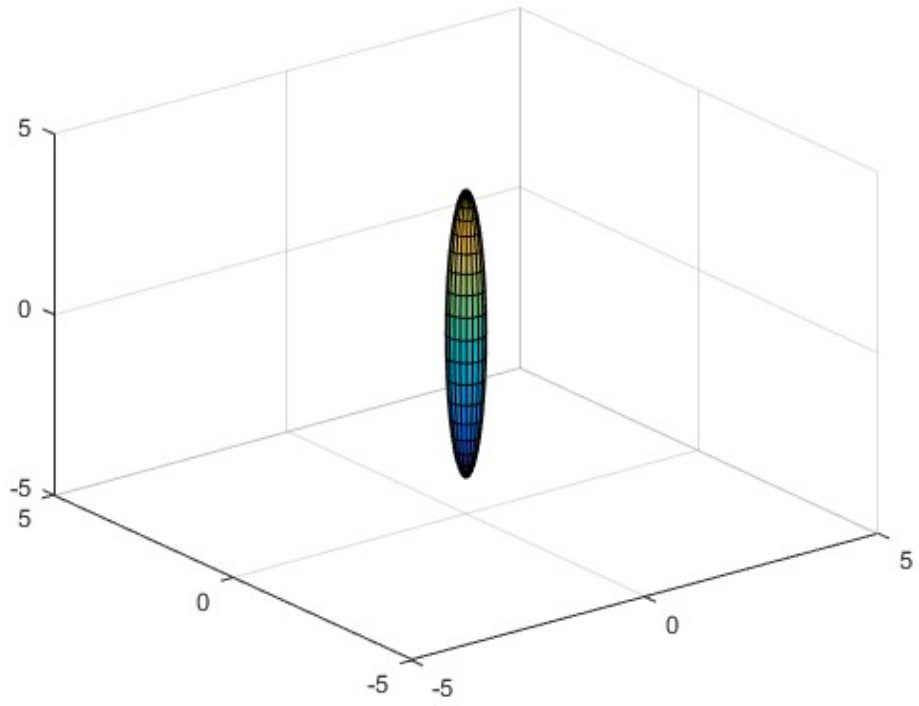
Строим графики функций

```
angles=0:20:180;
figure; % создаем новое окно

subplot(1,3,1), plot(angles,ZDR_crystals),
ylabel('Differential Reflectivity, dB');
% строим график зависимости ДО от угла сканирования антенны радиолокатора
grid on;
axis([0 180 -3 3])

subplot(1,3,2), plot(angles,LDR_crystals),
xlabel ('teta, deg'), ylabel('Linear Depolarization Ratio, dB');
% строим график зависимости ЛДО от угла сканирования антенны радиолокатора
grid on;
axis([0 180 -50 1]);

subplot(1,3,3), plot(angles,Rho_crystals);
ylabel('Correlation Coefficient, dB');
% строим график зависимости КК от угла сканирования антенны радиолокатора
grid on;
axis([0 180 0.9 1.1]);
```



Рисуем внешний вид эллипсоида, эмулирующего форму кристалла льда

```
figure;
a1=fun_a1(L);
a3=fun_a3(L);
[x,y,z] = ellipsoid(0,0,0,a1,a1,a3,20);
h=surf(x,y,z);
axis([-5 5 -5 5 -5 5]);
```



Дополнительные вызываемые функции:

```
function z=A1(L)
% эта функция вычисляет электромагнитные коэффициенты формы эллипсоида

global eps_r;
z=1./(1+lambda1(L).*(eps_r-1));
end

function z=A3(L)
% эта функция вычисляет электромагнитные коэффициенты формы эллипсоида

global eps_r;
z=1./(1+lambda3(L).*(eps_r-1));
end
```

```
function z=beta(L)
% эта функция определяет степень "сплюснутости" эллипсоида

z=fun_a1(L)./fun_a3(L);
end

function z=fun_a1(L)
% эта функция вычисляет длины двух второстепенных полуосей сфероида

z=fun_d(L)/2;
end

function z=fun_a3(L)
% эта функция вычисляет длину главной полуоси сфероида

z=L./2;
end

function z=fun_d(L)
% эта функция определяет зависимость диаметра частицы от длины

% для игольчатых кристаллов
% z=0.006*L.^0.52;

% для столбчатых кристаллов
z=0.1*L.^0.93;

% для пластинчатых кристаллов
% z=(100*L).^(1/0.42);

% for жидких капель
% z=L./(0.5*(exp(-(L.^2)/27))+0.5);
end

function sigma = fun_sigma(L)
% эта функция рассчитывает ЭПР эллипсоида согласно формуле Релея

global wavelength;
global eps_r;

sigma = (pi^5*L^6)/wavelength.^4*(((eps_r-1)/(eps_r+2))^2);
end

function z=illustrate(alpha,delta,phi,teta,L)
 % эта функция рисует эллипсоид с заданными соотношением сторон
```

```
 % и положением в пространстве

a1=fun_a1(L);
a3=fun_a3(L);
[x,y,z] = ellipsoid(0,0,0,a1,a1,a3,20);
h=surf(x,y,z);
% axis([-1 1 -1 1 -L L]);
center = [0 0 0];
dir1=[1 0 0];
rotate(h,dir1,radtodeg(delta),center);
dir2=[0 0 1];
rotate(h,dir2,radtodeg(alpha),center);
z=[phi_hh(alpha,delta,phi,teta) phi_vv(alpha,delta,phi,teta)];

function z=lambda1(L)
% эта функция рассчитывает коеффициент "лямбда", определяемый формой
% частицы

z=(1-lambda3(L))./2;

function z=lambda3(L)
% эта функция проверяет, является ли сфероид вытянутым или сплюснутым

if beta(L)<1&&beta(L)>0
     z=((1-E(L).^2)./(E(L).^2)).*(-1+(1/...
        (2.*E(L))).*log((1+E(L))./(1-E(L))));
    else
     z=((1+f(L).^2)/f(L).^2).*...
     (1-(1./f(L)).*atan(f(L)));
end


function z=E(L)
z=sqrt(1-beta(L).^2);


function z=f(L)

z=sqrt(beta(L).^2-1);

function Z=N_fun(L)
% эта функция определяет формулу распределения частиц по размерам
```

```
Z=1000.*L.^-2.3;
end

function z=phi_hh(alpha,delta,phi,teta)
% эта функция задает формулу 2.51 главы 2 диссертации

z=(sin(delta).^2).*(cos(alpha).^2).*(sin(phi).^2).*(sin(teta).^2)+...
  (sin(delta).^2).*(sin(alpha).^2).*(cos(phi).^2)+(cos(delta).^2).*...
  (sin(phi).^2).*(cos(teta).^2)-0.5.*sin(2.*delta).*cos(alpha).*...
  (sin(phi).^2).*sin(2.*teta)-0.5.*sin(2.*delta).*sin(alpha).*...
   sin(2.*phi).*cos(teta)+0.5.*(sin(delta).^2).*sin(2.*alpha).*...
   sin(2.*phi).*sin(teta);
end

function z=phi_vv(alpha,delta,phi,teta)
% эта функция задает формулу 2.52 главы 2 диссертации

z=(sin(delta).^2).*(cos(alpha).^2).*(cos(phi).^2).*(sin(teta).^2)+...
  (sin(delta).^2).*(sin(alpha).^2).*(sin(phi).^2)+(cos(delta).^2).*...
  (cos(phi).^2).*(cos(teta).^2)-0.5.*sin(2.*delta).*cos(alpha).*...
  (cos(phi).^2).*sin(2.*teta)+0.5.*sin(2.*delta).* sin(alpha).*...
  sin(2.*phi).*cos(teta)-0.5.*(sin(delta).^2).*sin(2.*alpha).*...
  sin(2.*phi).*sin(teta);
end

function z=phi_hv(alpha,delta,phi,teta)
% эта функция задает формулу 2.53 главы 2 диссертации

z=(0.5.*sin(2.*phi).*((sin(delta).^2).*(sin(alpha).^2)-(sin(delta).^2).*...
 (cos(alpha).^2).*(sin(teta).^2)-(cos(delta).^2)-sin(2.*delta).*...
  cos(alpha).*sin(2.*teta))-0.5.*cos(2.*phi).*((sin(2.*alpha).*...
 (sin(delta).^2).*sin(teta)-sin(2.*delta).*sin(alpha).*cos(teta))));

end

function [qhh,qvv,qhv]=q(alpha,delta,phi,teta,L)
% эта функция вычисляет относительную мощность принятого сигнала на разных
% поляризациях

qhh=((A1(L)+(A3(L)-A1(L)).*phi_hh(alpha,delta,phi,teta))).^2;
qvv=((A1(L)+(A3(L)-A1(L)).*phi_vv(alpha,delta,phi,teta))).^2;
qhv=((A3(L)-A1(L)).*phi_hv(alpha,delta,phi,teta)).^2;
end
```