Homework 3-NE 255

Information

Author: Mario I. Ortega
Organization: Department of Nuclear Engineering, University of California, Berkeley

1 Problem 1

We derive the SP_5 equation heuristically as follows: we can write the first-order from of the SP_N equation as

$$\nabla \cdot \vec{\phi_1} + \Sigma_a \phi_0 = s_0, \tag{1.1}$$

$$\left(\frac{l'+1}{2l'+1}\right)\nabla\phi_{l'+1} + \left(\frac{l'}{2l'+1}\right)\nabla\phi_{l'+1} + \Sigma_t\vec{\phi_{l'}} = \Sigma_{sl'}\vec{\phi_{l'}} + S_{l'}, \quad \text{for odd l'}, \tag{1.2}$$

$$\left(\frac{l'+1}{2l'+1}\right)\nabla\cdot\vec{\phi_{l'+1}} + \left(\frac{l'}{2l'+1}\right)\nabla\cdot\vec{\phi_{l'-1}} + \Sigma_t\phi_{l'} = \Sigma_{sl'}\phi_{l'} + S_{l'}, \quad \text{for even l' } > 0. \tag{1.3}$$

For SP_5, we obtain the following set of equations

$$\nabla \cdot \vec{\phi_1} + \Sigma_a \phi_0 = s_0, \tag{1.4}$$

$$\frac{2}{3}\nabla\phi_2 + \frac{1}{3}\nabla\phi_0 + \Sigma_t\vec{\phi_1} = \Sigma_{s1}\vec{\phi_1} + s_1, \tag{1.5}$$

$$\frac{3}{5}\nabla\cdot\vec{\phi_3} + \frac{2}{5}\nabla\cdot\vec{\phi_1} + \Sigma_t\phi_2 = \Sigma_{s2}\phi_2 + s_2, \tag{1.6}$$

$$\frac{4}{7}\nabla\phi_4 + \frac{3}{7}\nabla\phi_2 + \Sigma_t\vec{\phi_3} = \Sigma_{s1}\vec{\phi_3} + s_3, \tag{1.7}$$

$$\frac{5}{9}\nabla\cdot\vec{\phi_5} + \frac{4}{9}\nabla\cdot\vec{\phi_3} + \Sigma_t\phi_4 = \Sigma_{s4}\phi_4 + s_4, \tag{1.8}$$

$$\frac{6}{11}\nabla\phi_6 + \frac{5}{11}\nabla\phi_4 + \Sigma_t\vec{\phi_5} = \Sigma_{s5}\vec{\phi_5} + s_5. \tag{1.9}$$

We set $\phi_6 = 0$ in the SP_5 approximation and assume an isotropic source to obtain the following SP_5 equations:

$$\nabla \cdot \vec{\phi_1} + \Sigma_a \phi_0 = s_0, \tag{1.10}$$

$$\frac{2}{3}\nabla\phi_2 + \frac{1}{3}\nabla\phi_0 + \Sigma_t\vec{\phi_1} = \Sigma_{s1}\vec{\phi_1}, \tag{1.11}$$

$$\frac{3}{5}\nabla\cdot\vec{\phi_3} + \frac{2}{5}\nabla\cdot\vec{\phi_1} + \Sigma_t\phi_2 = \Sigma_{s2}\phi_2, \tag{1.12}$$

$$\frac{4}{7}\nabla\phi_4 + \frac{3}{7}\nabla\phi_2 + \Sigma_t\vec{\phi_3} = \Sigma_{s1}\vec{\phi_3}, \tag{1.13}$$

$$\frac{5}{9}\nabla\cdot\vec{\phi_5}+\frac{4}{9}\nabla\cdot\vec{\phi_3}+\Sigma_t\phi_4=\Sigma_{s4}\phi_4, \tag{1.14}$$

$$\frac{5}{11}\nabla\phi_4+\Sigma_t\vec{\phi_5}=\Sigma_{s5}\vec{\phi_5}. \tag{1.15}$$

We apply the Marshak boundary conditions by replacing the ϕ_n with the SP_N unknowns and mu with $\hat{n}\cdot\hat{\Omega}$ where $\hat{n}$ is the unit inward normal to the boundary to obtain the following expression for vacuum boundary conditions:

$$\sum_{n=0,2,4}^{N}\frac{2n+1}{4\pi}\phi_n\int_{\hat{n}\cdot\hat{\Omega}}P_{2m-1}(\hat{n}\cdot\hat{\Omega})d\hat{\Omega}+\sum_{n=1,3,5}^{N}\frac{2n+1}{4\pi}\hat{n}\cdot\vec{\phi_n}\int_{\hat{n}\cdot\hat{\Omega}}P_{2m-1}(\hat{n}\cdot\hat{\Omega})d\hat{\Omega}=0. \tag{1.16}$$

2 Problem 2

Consider the integral

$$\int_{4\pi}d\hat{\Omega}|\hat{\Omega}|,$$

using the LQ_N quadrature, we evaluate the integral as follows:

2.1 (a)

Using the S_4 LQ_N quadrature set shown in Figure **??**, we evaluate the integral using the fact

$$|\hat{\Omega}|=\sqrt{\mu^2+\eta^2+\xi^2}=1.$$

There are three ordinates per quadrants whose value is 1. The quadrature weight for each ordinate is 1/3 so the quadrature sum is equal to 1 in each octant of the unit sphere. Repeating this calculation for each octant we obtain

$$\sum_{i=1}^{24}w_i f(\Omega_i)=8$$

where we have 24 ordinates given by the relation $N(N+2)$ where N is the S_N order. The quadrature sum is normalized using the fact

$$\int_{4\pi}d\hat{\Omega}=4\pi$$

by the normalization factor $4\pi/8$ so that our integral for S_4 is given as

$$I_4=\frac{4\pi}{8}\sum_{i=1}^{24}w_i\Omega_i=1.$$

2.2 (b)

Using the S_6 LQ_N quadrature set shown in Figure **??**, we evaluate the integral now using 48 directions ($N=6$). In each octant, there are six ordinates each with a weight of either w = 0.1761263 or 0.1572071. The corresponding weight of the ordinate is given by the location on the unit sphere octant. Using this, once again, the sum is one is each octant for a total of 8 over the whole unit sphere. Multiplying by the normalization constant, we obtain

$$I_4=\frac{4\pi}{8}\sum_{i=1}^{24}w_i\Omega_i=1.$$

For both the S_4 and the S_6 quadrature sets, we obtain the same solution as expected.

2.3 (c)

Two codes were written to implement the quadrature and can be seen below. One function returns the ordinate points and their corresponding weights for both the S_4 and S_6 cases. The other function evaluates a function handle at the ordinate points and sums up the ordinates with their corresponding weight to determine the value of the integral over the unit sphere. A few functions were tested and the script can be seen after this section. For all cases, the quadrature computed the integral correctly.

- $f(\hat{\Omega}) = 1$

$$\int_{4\pi} d\hat{\Omega} = 4\pi, \quad \frac{4\pi}{8}\sum_{i=1}^{24} w_i = 4\pi, \quad \frac{4\pi}{8}\sum_{i=1}^{48} w_i = 4\pi \tag{2.1}$$

- $f(\hat{\Omega}) = \mu$

$$\int_{4\pi} \mu d\hat{\Omega} = 0, \quad \frac{4\pi}{8}\sum_{i=1}^{24} w_i f(\Omega_i) = 0, \quad \frac{4\pi}{8}\sum_{i=1}^{48} w_i f(\Omega_i) = 0 \tag{2.2}$$

- $f(\hat{\Omega}) = \mu^2$

$$\int_{4\pi} \mu^2 d\hat{\Omega} = \frac{4\pi}{3}, \quad \frac{4\pi}{8}\sum_{i=1}^{24} w_i f(\Omega_i) = 4.188790, \quad \frac{4\pi}{8}\sum_{i=1}^{48} w_i f(\Omega_i) = 4.188790 \tag{2.3}$$

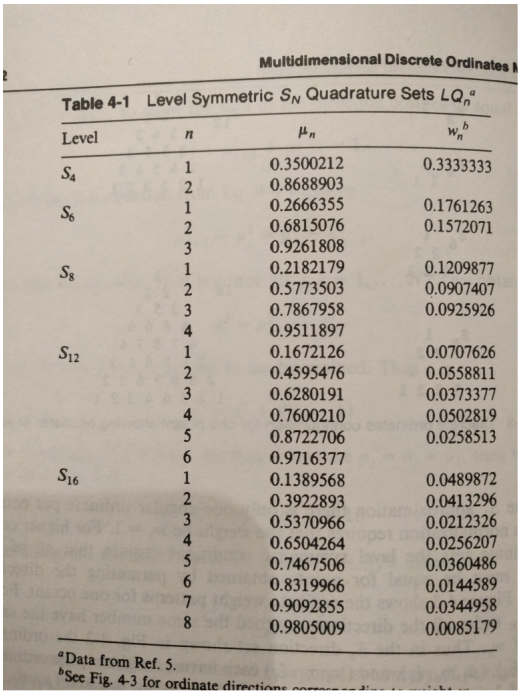
2.4 Problem 2 MATLAB Functions and Scripts

```
function [dirs,wi] = LQnQuadrature(SN)

if ( SN == 4 )

    mu(1) = 0.3500212;
    mu(2) = 0.8688903;
    mu(3) = -mu(2);
    mu(4) = -mu(1);

    eta = mu;
    xi = mu;

    ordinates = combvec(mu,eta,xi);
    mag = round(sum(ordinates.^2),6);

    locs = mag == 1;

    dirs = ordinates(:,locs);
    wi = 0.3333333.*ones(1,length(dirs(1,:)));

elseif ( SN == 6 )

    mu(1) = 0.2666355;
    mu(2) = 0.6815076;
    mu(3) = 0.9261808;
    mu(4) = -mu(3);
    mu(5) = -mu(2);
    mu(6) = -mu(1);

    eta = mu;
    xi = mu;

    w(1) = 0.1761263;
    w(2) = 0.1572071;

    ordinates = combvec(mu,eta,xi);
    mag = round(sum(ordinates.^2),6);

```

```
    locs = mag == 1;

    dirs = ordinates(:,locs);

    for i = 1:length(dirs)

        if ( abs(dirs(3,i)) == max(xi) )

            dirs(4,i) = w(1);

        elseif ( abs(dirs(3,i)) == median(abs(xi)) )

            dirs(4,i) = w(2);

        elseif ( abs(dirs(3,i)) == min(abs(xi)) && abs(dirs(2,i)) == median(abs(eta)) &&...
                abs(dirs(1,i)) == median(abs(mu)) )

            dirs(4,i) = w(2);

        else

            dirs(4,i) = w(1);

        end

    end

    wi = dirs(4,:);
    dirs = dirs(1:3,:);

else

    print('Higher Order LQn Sets to be Implemented')

end


return
```

```
function I = LQnQuadratureIntegration(f,N)

[dirs,w] = LQnQuadrature(N);

sumI = 0;

for i = 1:length(w)

    mu = dirs(1,i);
    eta = dirs(2,i);
    xi = dirs(3,i);

    sumI = sumI + w(i).*f(mu,eta,xi);

end

I = sumI*(4*pi/8);

return
```

```
%NE255
%Problem 3c

clc, clear

```



Multidimensional Discrete Ordinates

Table 4-1 Level Symmetric S_N Quadrature Sets LQ_n[a]

Level	n	μ_n	w_n[b]
S_4	1	0.3500212	0.3333333
	2	0.8688903	
S_6	1	0.2666355	0.1761263
	2	0.6815076	0.1572071
	3	0.9261808	
S_8	1	0.2182179	0.1209877
	2	0.5773503	0.0907407
	3	0.7867958	0.0925926
	4	0.9511897	
S_{12}	1	0.1672126	0.0707626
	2	0.4595476	0.0558811
	3	0.6280191	0.0373377
	4	0.7600210	0.0502819
	5	0.8722706	0.0258513
	6	0.9716377	
S_{16}	1	0.1389568	0.0489872
	2	0.3922893	0.0413296
	3	0.5370966	0.0212326
	4	0.6504264	0.0256207
	5	0.7467506	0.0360486
	6	0.8319966	0.0144589
	7	0.9092855	0.0344958
	8	0.9805009	0.0085179

[a]Data from Ref. 5.
[b]See Fig. 4-3 for ordinate directions

Figure 2.1: LQ_N Quadrature

```
%f(Omega) = 1
f = @(x,y,z) 1;
I4 = LQnQuadratureIntegration(f,4);
I6 = LQnQuadratureIntegration(f,6);

fprintf('S4 Quadrature Result: %f S6 Quadrature Result: %f \n',I4,I6);

%Result
%S4 Quadrature Result: 12.566369 S6 Quadrature Result: 12.566373

%f(Omega) = mu
f = @(x,y,z) x;
I4 = LQnQuadratureIntegration(f,4);
I6 = LQnQuadratureIntegration(f,6);

fprintf('S4 Quadrature Result: %f S6 Quadrature Result: %f \n',I4,I6);

%Result
%S4 Quadrature Result: 0.000000 S6 Quadrature Result: 0.000000

%f(Omega) = mu^2
f = @(x,y,z) x^2;
I4 = LQnQuadratureIntegration(f,4);
I6 = LQnQuadratureIntegration(f,6);

fprintf('S4 Quadrature Result: %f S6 Quadrature Result: %f \n',I4,I6);

%Result
%S4 Quadrature Result: 4.188790 S6 Quadrature Result: 4.188790
```

3 Problem 3

3.1 (a)

- The Diffusion Equation
  - The diffusion equation is applicable when scattering is mostly isotropic (weakly linearly anisotropic) and far away from boundaries, strong absorbers, and material discontinuities. The diffusion equation is simple to solve as it is a well studied partial differential equation. It is cheaper to solve than the transport equation as it removed the angular dependence of the transport equation.
- Deterministic Methods
  - Deterministic methods are applicable if you can accurately discretize your problem in space, energy, and angle. They are fast since linear algebra algorithms are fast, they produce global solutions, but require multigroup cross sections. Deterministic methods can be used to solve both the diffusion equation and the transport equation. Deterministic methods are faster than Monte Carlo methods at the cost of discretization error in space, angle, and energy and are somewhat more complicated mathematically.
- Monte Carlo Methods
  - Monte Carlo methods are always applicable as they directly simulate the physics of neutron transport. They do not require discretization of any kind and use probability distribution functions to obtain the mean behavior of a system. Monte Carlo methods are easy to parallelize but require billions of particle histories to determine average quantities and reduce statistical error. Monte Carlo methods can be used to determine quantities on complex geometries but require the use of variance reduction techniques at times. Monte Carlo methods are much more costly than deterministic methods.

3.2 (b)

Some of the advantages of deterministic methods are that they are fast, give global solutions, and do not introduce statistical noise due to random sampling. However, some disadvantages are that they introduce discretization error, must use multigroup cross sections, and the algorithms are difficult to parallelize.

4 Problem 4

4.1 (a)

Spherical harmonics have been plotted for $\ell = 0, 1, 2$ with each having an m for $-\ell \leq m \leq \ell$ in Figures 4.1, 4.2, and 4.3.

4.2 (b)

The external source was approximated using S_4 quadrature and $q_e = 1$ for all angles for $\ell = 0, 1, 2$. The external source was expanded using the following relation

$$q_e^g(\vec{r}, \hat{\Omega}) = \sum_{l=0}^{N} \left[Y_{l0}^e(\hat{\Omega}) q_{l0}^g(\vec{r}) + \sum_{m=1}^{l} \left(Y_{lm}^e(\hat{\Omega}) q_{lm}^g(\vec{r}) + Y_{lm}^o(\hat{\Omega}) s_{lm}^g(\vec{r}) \right) \right], \tag{4.1}$$

where

$$q_{lm}^g = \int_{4\pi} Y_{lm}^e(\hat{\Omega}) q_e^g d\hat{\Omega}, \quad m \geq 0, \tag{4.2}$$

$$s_{lm}^g = \int_{4\pi} Y_{lm}^o(\hat{\Omega}) q_e^g d\hat{\Omega}, \quad m > 0, \tag{4.3}$$

where

$$Y_{lm}^{e} = D_{lm} P_{lm} \cos(m\varphi), \tag{4.4}$$

$$Y_{lm}^{e=o} = D_{lm} P_{lm} \sin(m\varphi), \tag{4.5}$$

$$D_{lm} = (-1)^m \sqrt{(2-\delta_{m0}) \frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}}. \tag{4.6}$$

The external source was plotted for the various l and can be seen in Figure 4.4. The MATLAB functions and scripts to calculate the external source can be seen below.

4.3 Problem 4 MATLAB Functions and Scripts

```
%NE255
%Problem 4b

theta = linspace(0,pi,30);
phi = linspace(0,2*pi,30);

[theta,phi] = meshgrid(theta,phi);

%l = 0
l = [0,1,2];

for k = 1:length(l)

    clf

qsum = 0;

for i = 0:l(k)

    f = Ylme(i,0);

    qsum = qsum + f(theta,phi).*qlm(i,0,1);

    for m = 1:i

        g = Ylme(i,m);
        h = Ylmo(i,m);

        qsum = qsum + g(theta,phi)*q(i,m,1) + h(theta,phi)*q(i,m,1);

    end

end

q = qsum;

x = q.*sin(theta).*cos(phi);
y = q.*sin(theta).*sin(phi);
z = q.*cos(theta);

filename = sprintf('extern_src_ell_%i',l(k));

figure(1)
surf(x,y,z)
axis equal
xlabel('x')
ylabel('y')
str = sprintf('External Source, l = %i \n',l(k));
title(str,'FontSize',16);
```

```
50
51 export_fig(filename,'-pdf','-nocrop');
52
53 end
```

```
1  function Ylme = Ylme(l,m)
2
3  %Generates Even Spherical Harmonic for l,m
4
5  if ( m == 0 )
6
7      dlm = ((-1)^m)*sqrt(((2*l+1)/(4*pi))*(factorial(l-m)/factorial(l+m)));
8
9  else
10
11     dlm = ((-1)^m)*sqrt(2*((2*l+1)/(4*pi))*(factorial(l-m)/factorial(l+m)));
12
13 end
14
15 plm = AssociatedLegendrePolynomial(l,m);
16
17 Ylme = @(theta,phi) dlm.*plm(cos(theta)).*cos(m.*phi);
18
19 return
```

```
1  function Ylmo = Ylmo(l,m)
2
3  %Generates Odd Spherical Harmonic for l,m
4
5  if ( m == 0 )
6
7      dlm = ((-1)^m)*sqrt(((2*l+1)/(4*pi))*(factorial(l-m)/factorial(l+m)));
8
9  else
10
11     dlm = ((-1)^m)*sqrt(2*((2*l+1)/(4*pi))*(factorial(l-m)/factorial(l+m)));
12
13 end
14
15 plm = AssociatedLegendrePolynomial(l,m);
16
17 Ylmo = @(theta,phi) dlm.*plm(cos(theta)).*sin(m.*phi);
18
19 return
```

```
1  function qlm = qlm(l,m,q)
2
3  %Generates Even Source Moment for l,m
4
5  if ( m == 0 )
6
7      dlm = ((-1)^m)*sqrt(((2*l+1)/(4*pi))*(factorial(l-m)/factorial(l+m)));
8
9  else
10
11     dlm = ((-1)^m)*sqrt(2*((2*l+1)/(4*pi))*(factorial(l-m)/factorial(l+m)));
12
13 end
14
15 plm = AssociatedLegendrePolynomial(l,m);
16
17 qlmg = @(theta,phi) dlm.*plm(cos(theta)).*cos(m.*phi).*q;
```

```
18
19 qlm = LQnQuad(qlmg,4);
20
21 return
```

```
1  function slm = slm(l,m,q)
2
3  %Generates Odd Source Moment for l,m
4
5  if ( m == 0 )
6
7      dlm = ((-1)^m)*sqrt(((2*l+1)/(4*pi))*(factorial(l-m)/factorial(l+m)));
8
9  else
10
11     dlm = ((-1)^m)*sqrt(2*((2*l+1)/(4*pi))*(factorial(l-m)/factorial(l+m)));
12
13 end
14
15 plm = AssociatedLegendrePolynomial(l,m);
16
17 slmg = @(theta,phi) dlm.*plm(cos(theta)).*sin(m.*phi).*q;
18
19 slm = LQnQuad(slmg,4);
20
21 return
```

Problem 5

The major nuclear data libraries and the countries that manage them are

- ENDF (Evaluated Nuclear Data File) - United States of America
- JEFF (Joint Fission Fusion File) - European Union
- JENDL (Japanese Evaluated Nuclear Data Library) - Japan
- CENDL (Chinese Evaluated Nuclear Data Library) - People's Republic of China
- BROND - Russian Federation.

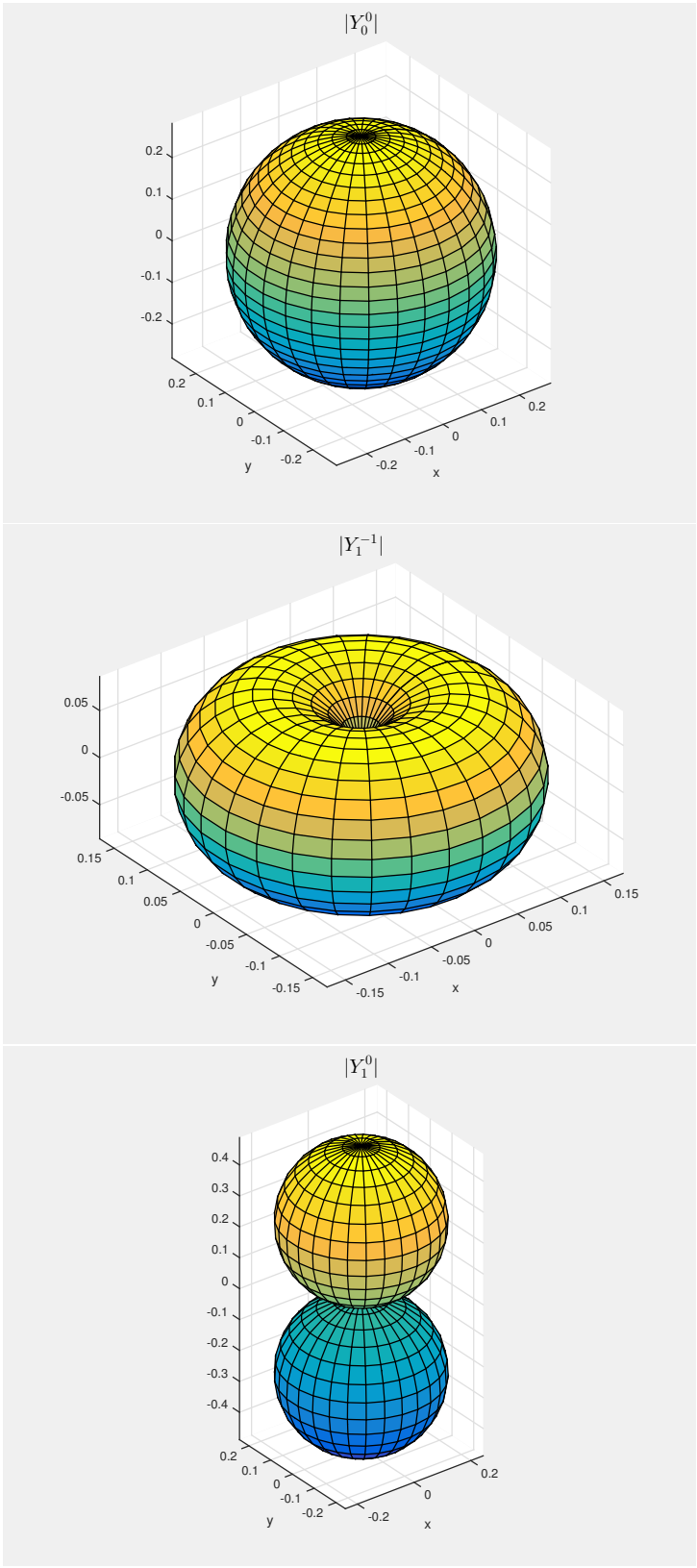


Figure 4.1: Spherical Harmonics

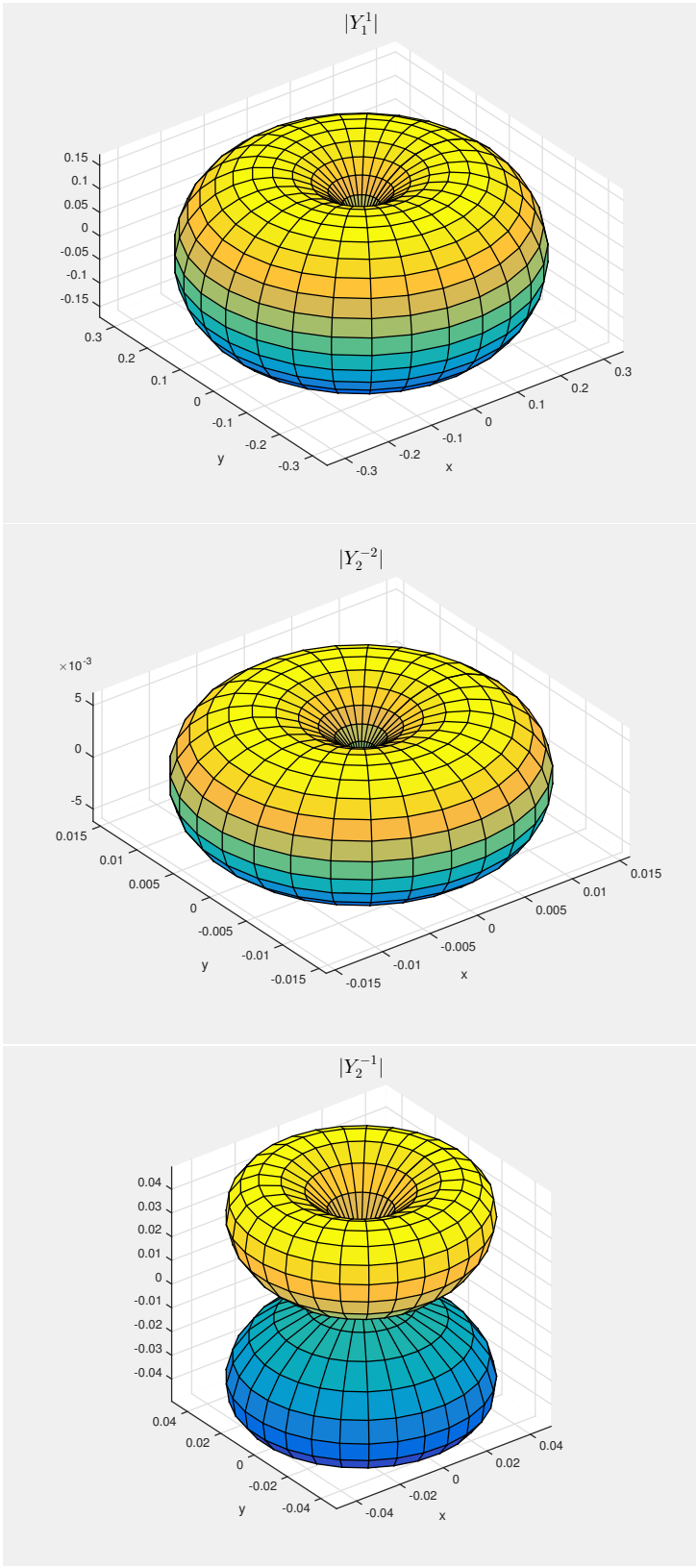


Figure 4.2: Spherical Harmonics

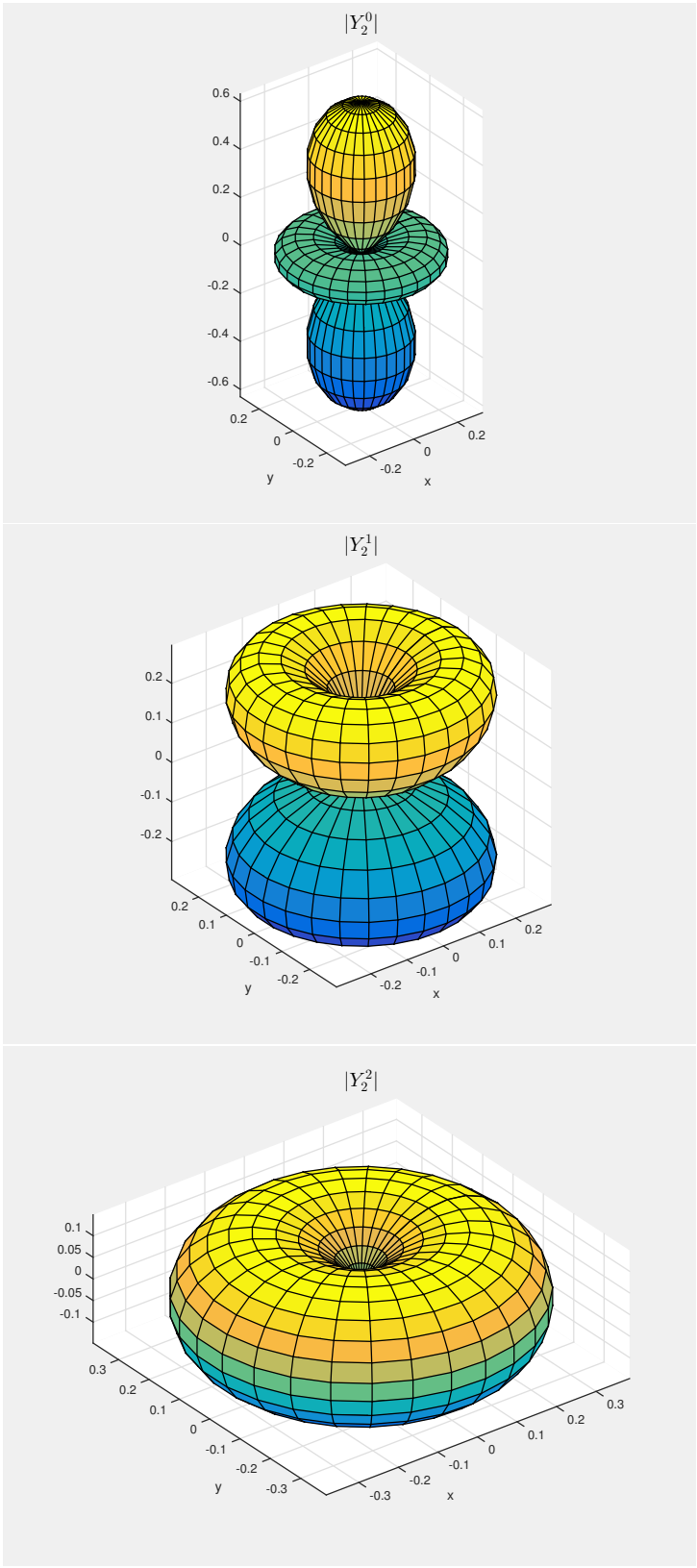


Figure 4.3: Spherical Harmonics

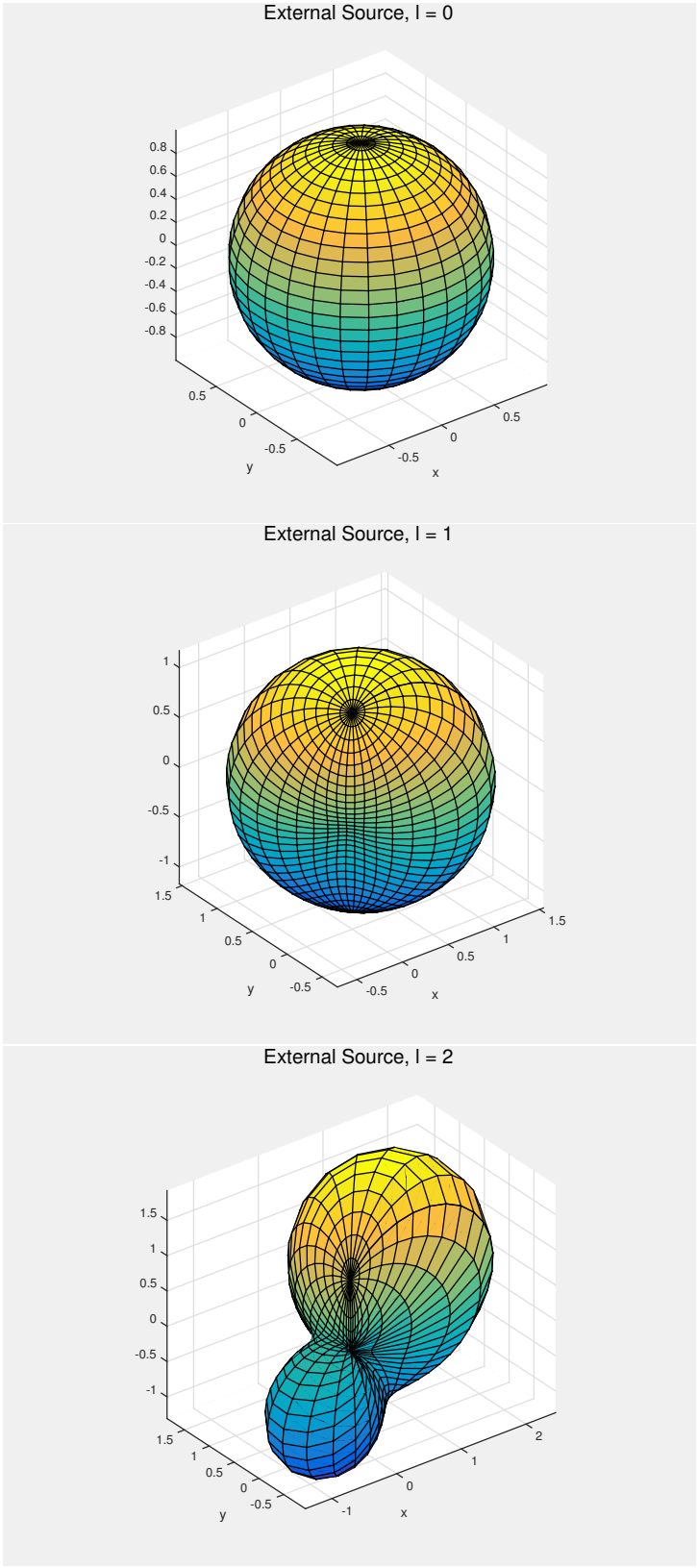


Figure 4.4: External Source Expansion