
Table of Contents

.....	1
Omega 1	1
Omega 2	3
Omega 3	5

```
clear all
close all
clc
```

Omega 1

```
figure('Name', 'Part 1')

% plotting a sphere in MATLAB requires converting phi, theta and r into x,
% y and z coordinates.

%linspace is a function that produces a set number of points between two
%numbers, in this case 0 and 100

theta=linspace(0,pi/2,100); %defining angles for theta
phi=linspace(0,pi/2,100); %degining angles for phi

%meshgrid produces an n x m matrix x points placed horizontally along the
%matrix and y points placed vertically along the matrix

[theta,phi]=meshgrid(theta,phi); %meshgrid converts coordinates into table
format

%the radius is...
rho=1;

%defining x, y and z coordinates
x1=rho*sin(phi).*cos(theta);
y1=rho*sin(phi).*sin(theta);
z1=rho*cos(phi);

%parameterizations - yes, there's a way to do this on MATLAB :)
syms theta_sym phi_sym

%the paramaterized surface is defined as ...
f=[sin(phi_sym)*cos(theta_sym),sin(phi_sym)*sin(theta_sym),cos(phi_sym)];

%the partial diff of the surface f with respect to phi and theta
dfdphi=diff(f,phi_sym);
dfdtheta=diff(f,theta_sym);
```

```

%taking the cross product of both partial diffs
crossP1=cross(dfdphi,dfdtheta);

surf(x1,y1,z1);
hold on;

%labelling axis coordinates
xlabel('x')
ylabel('y')
zlabel('z')

%adding a title to the plot
title('P1- Segment derived from sphere (x^2+y^2+z^2=1)')
% Normal Vectors
% define the surface parameterisation
rx = @(u,v) sin(u).*cos(v);
ry = @(u,v) sin(u).*sin(v);
rz = @(u,v) cos(u);
% define the normal vector field u = phi, v = theta
Nx = @(u,v) cos(v).*sin(u).^2;
Ny = @(u,v) sin(u).^2.*sin(v);
Nz = @(u,v) cos(u).*sin(u);

[U,V] = meshgrid(0:0.2:1.5,0:0.2:1.5);
quiver3(rx(U,V),ry(U,V),rz(U,V),Nx(U,V),Ny(U,V),Nz(U,V),'k')

Fx = @(u,v) sin(u).*cos(v);
Fy = @(u,v) sin(u).*sin(v);
Fz = @(u,v) 5*ones(size(u));

% Plot the vector field
quiver3(rx(U,V),ry(U,V),rz(U,V),Fx(U,V),Fy(U,V),Fz(U,V),'Color','m')
legend("P1", "Normal Vectors", "Field Vectors")
hold off;
%the parameterized vector field
F= [sin(phi_sym)*cos(theta_sym),sin(phi_sym)*sin(theta_sym),5];

% taking the dot product between F and the cross product found before
dotP1=dot(F,crossP1);

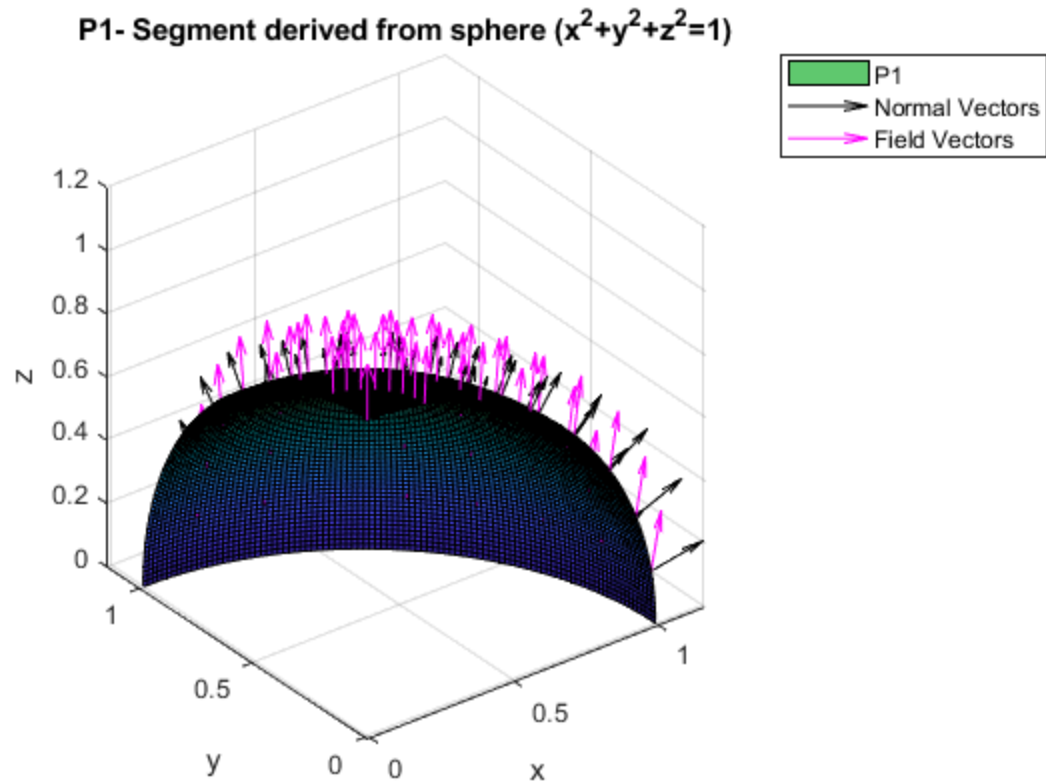
%finding the double integral
int1 = int(int(dotP1, phi_sym, 0, pi/2), theta_sym, 0, pi/2);

% representing the final answer in decimal form
final_answer_P1=double(int1);

fprintf(' The final answer (for P1) is: %.2f\n',final_answer_P1)

The final answer (for P1) is: 4.97

```



Omega 2

```
figure('Name', 'Part 2')
```

```
% plotting a sphere in MATLAB requires converting phi, theta and r into x,
% y and z coordinates.
```

```
%linspace is a function that produces a set number of points between two
%numbers, in this case 0 and 100
```

```
theta=linspace(0,pi/2,100); %defining angles for theta
r=linspace(0,1,100); %degining angles for phi
```

```
%meshgrid produces an n x m matrix x points placed horizontally along the
%matrix and y points placed vertically along the matrix
```

```
[theta,r]=meshgrid(theta,r); %meshgrid converts coordinates into table format
```

```
%defining x, y and z coordinates
x1=r.*cos(theta);
y1=r.*sin(theta);
z1=1 - r.^2;
```

```
%parameterizations - yes, there's a way to do this on MATLAB :)
syms theta_sym rad
```

```

%the parameterized surface is defined as ...
f=[rad.*cos(theta_sym),rad.*sin(theta_sym),1-rad.^2];

%the partial diff of the surface f with respect to phi and theta
dfdrad=diff(f,rad);
dfdtheta=diff(f,theta_sym);

%taking the cross product of both partial diffs
crossP2=cross(dfdrad,dfdtheta);
surf(x1,y1,z1);
hold on;
%labelling axis coordinates
xlabel('x')
ylabel('y')
zlabel('z')

%adding a title to the plot
title('P2- Segment derived from sphere (z = 1 - x^2 - y^2)')

% Normal Vectors
% define the surface parameterisation
rx = @(u,v) u.*cos(v);
ry = @(u,v) u.*sin(v);
rz = @(u,v) 1-u.^2;
% define the normal vector field u = phi, v = theta
Nx = @(u,v) 2.*u.^2.*cos(v);
Ny = @(u,v) 2.*u.^2.*sin(v);
Nz = @(u,v) u;

[U,V] = meshgrid(0:0.2:1,0:0.2:1.55);
quiver3(rx(U,V),ry(U,V),rz(U,V),Nx(U,V),Ny(U,V),Nz(U,V),'r')

% Plot the vector field
quiver3(rx(U,V),ry(U,V),rz(U,V),Fx(U,V),Fy(U,V),Fz(U,V),'Color','m')
legend("P2", "Normal Vectors", "Field Vectors")

hold off;
%the parameterized vector field
F= [rad.*cos(theta_sym),rad.*sin(theta_sym),5];

% taking the dot product between F and the cross product found before
dotP2=dot(F,crossP2);

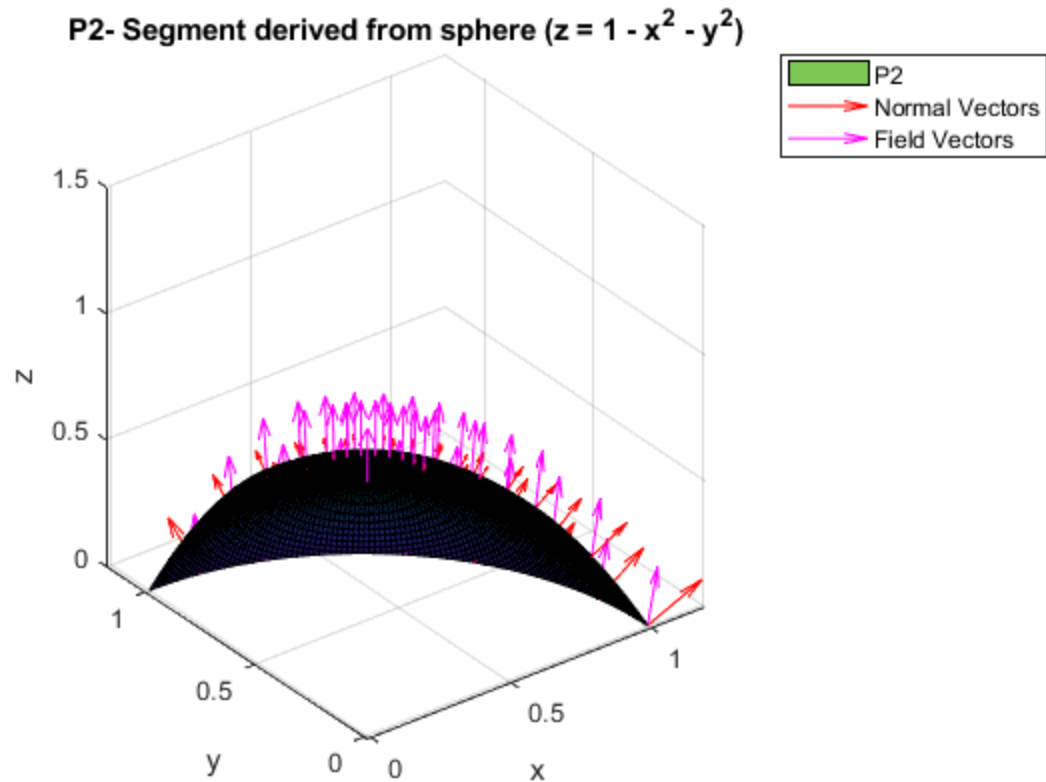
%finding the double integral
int1 = int(int(dotP2, theta_sym, 0, pi/2), rad, 0, 1);

% representing the final answer in decimal form
final_answer_P2=double(int1);

fprintf(' The final answer (for P2) is: %.2f\n',final_answer_P2)

The final answer (for P2) is: 4.71

```



Omega 3

```
figure('Name', 'Part 3')
```

```
% plotting a sphere in MATLAB requires converting phi, theta and r into x,
% y and z coordinates.
```

```
%linspace is a function that produces a set number of points between two
%numbers, in this case 0 and 100
```

```
theta=linspace(0,pi/2,100); %defining angles for theta
r=linspace(0,1,100); %degining angles for phi
```

```
%meshgrid produces an n x m matrix x points placed horizontally along the
%matrix and y points placed vertically along the matrix
```

```
[theta,r]=meshgrid(theta,r); %meshgrid converts coordinates into table format
```

```
%the radius is...
```

```
%defining x, y and z coordinates
x1=r.*cos(theta);
y1=r.*sin(theta);
z1=1 - r;
```

```

%parameterizations - yes, there's a way to do this on MATLAB :)
syms theta_sym rad

%the parameterized surface is defined as ...
f=[rad.*cos(theta_sym),rad.*sin(theta_sym),1-rad];

%the partial diff of the surface f with respect to phi and theta
dfdtrad=diff(f,rad);
dfdtheta=diff(f,theta_sym);

%taking the cross product of both partial diffs
crossP3=cross(dfdtrad,dfdtheta);
surf(xl,y1,z1);
hold on;
%labelling axis coordinates
xlabel('x')
ylabel('y')
zlabel('z')

%adding a title to the plot
title('P3- Segment derived from Cone (z = 1 - sqrt(x^2 + y^2))')
% Normal Vectors
% define the surface parameterisation
rx = @(u,v) u.*cos(v);
ry = @(u,v) u.*sin(v);
rz = @(u,v) 1-u;
% define the normal vector field u = phi, v = theta
Nx = @(u,v) 2.*u.*cos(v);
Ny = @(u,v) 2.*u.*sin(v);
Nz = @(u,v) u;

[U,V] = meshgrid(0:0.2:1,0:0.2:1.55);
quiver3(rx(U,V),ry(U,V),rz(U,V),Nx(U,V),Ny(U,V),Nz(U,V),'r')

% Plot the vector field
quiver3(rx(U,V),ry(U,V),rz(U,V),Fx(U,V),Fy(U,V),Fz(U,V),'Color','m')
legend("P3", "Normal Vectors", "Field Vectors")

hold off;
%the parameterized vector field
F= [rad.*cos(theta_sym),rad.*sin(theta_sym),5];

% taking the dot product between F and the cross product found before
dotP3=dot(F,crossP3);

%finding the double integral
int1 = int(int(dotP3, theta_sym, 0, pi/2), rad, 0, 1);

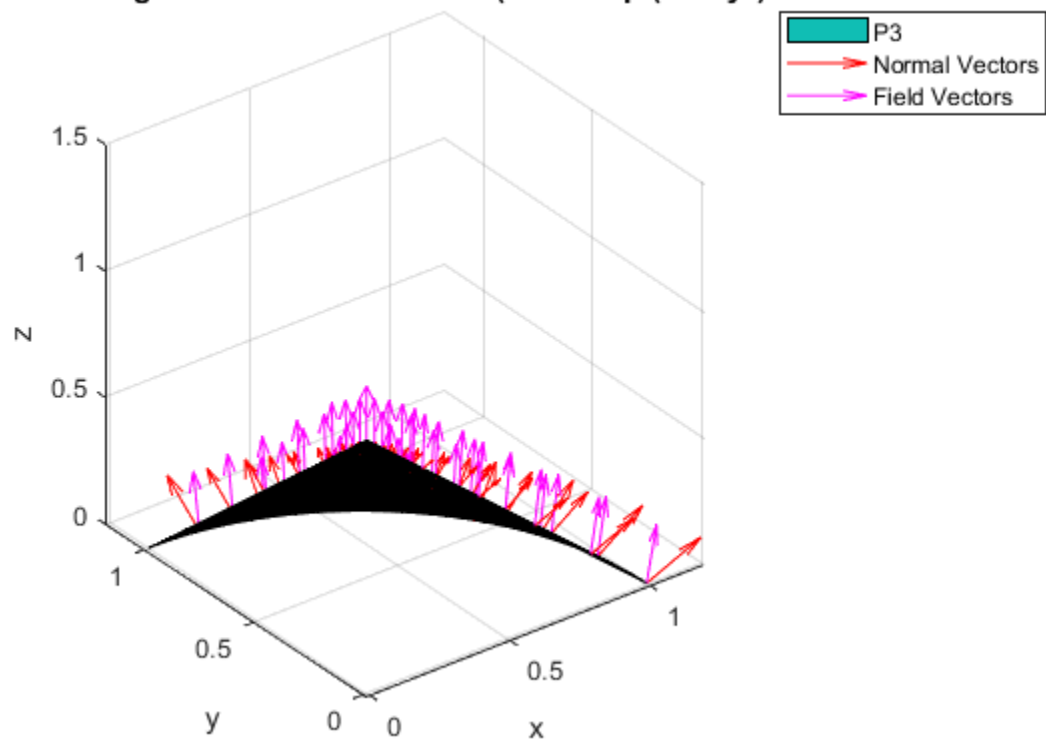
% representing the final answer in decimal form
final_answer_P3=double(int1);

fprintf(' The final answer (for P3) is: %.2f\n',final_answer_P3)

```

The final answer (for P3) is: 4.45

P3- Segment derived from Cone ($z = 1 - \sqrt{x^2 + y^2}$)



Published with MATLAB® R2023a