Title: Evaluating Volume under Surfaces

Aim: To find the volume under the given surfaces using the double integral method.

Question:

Example Problems

1. Set up a double integral to find the volume of a sphere of unit radius.
2. Find the volume of the solid that lies under the paraboloid $z = x^2 + y^2$ and above the region D in the xy-plane bounded by the lines $y = 2x$ and the parabola $y = x^2$.
3. Evaluate $\iint_R (x - 3y^2)\,dA$ where $R = \{(x, y) \mid 0 \le x \le 2, 1 \le y \le 2\}$
4. Evaluate $\iint_R y\sin(xy)\,dA$ where $R = [1, 2] \times [0, \pi]$

Converting Cartesian to polar coordinates:

5. Find the volume of the solid bounded by the plane z=0 and the paraboloid $z = 1 - x^2 - y^2$.
6. Find the volume of the solid that lies under the cone $z = x^2 + y^2$ and above the xy-plane, and inside the cylinder $x^2 + y^2 = 2x$.

Exercise Problems:

1. Set up a double integral to find the volume of the *hoof of Archimedes*, which is the solid region bounded by the planes $z = y$, $z = 0$, and the cylinder $x^2 + y^2 = 1$.
2. Write an iterated integral to view the volume enclosed by the cone $z^2 = x^2 + y^2$ and above the disk $x^2 + y^2 \le 1$. Hence find the volume .
3. Find the volume of the solid bounded by the elliptic paraboloid $x^2 + 2y^2 + z = 16$, the planes $x = 2$ and $y = 2$ and the three coordinate planes.
4. Use polar coordinates to find the volume of the solid that lies under the cone $z = \sqrt{x^2 + y^2}$ and above the disk $x^2 + y^2 \le 4$.

FUNCTION viewSolid.m:

```
function viewSolid(zvar, F, G, yvar, f, g, xvar, a, b)
%VIEWSOLID is a version for MATLAB of the routine on page 161
% of "Multivariable Calculus and Mathematica" for viewing the region
% bounded by two surfaces for the purpose of setting up triple integrals.
% The arguments are entered from the inside out.
% There are two forms of the command --- either f, g,
% F, and G can be vectorized functions, or else they can
% be symbolic expressions. xvar, yvar, and zvar can be
% either symbolic variables or strings.
% The variable xvar (x, for example) is on the
% OUTSIDE of the triple integral, and goes between CONSTANT limits a and
b.
% The variable yvar goes in the MIDDLE of the triple integral, and goes
% between limits which must be expressions in one variable [xvar].
% The variable zvar goes in the INSIDE of the triple integral, and goes
% between limits which must be expressions in two
% variables [xvar and yvar]. The lower surface is plotted in red, the
% upper one in blue, and the "hatching" in cyan.
%
% Examples: viewSolid(z, 0, (x+y)/4, y, x/2, x, x, 1, 2)
% gives the picture on page 163 of "Multivariable Calculus and
Mathematica"
% and the picture on page 164 of "Multivariable Calculus and Mathematica"
% can be produced by
% viewSolid(z, x^2+3*y^2, 4-y^2, y, -sqrt(4-x^2)/2, sqrt(4-x^2)/2, ...
% x, -2, 2,)
% One can also type viewSolid('z', @(x,y) 0, ...
% @(x,y)(x+y)/4, 'y', @(x) x/2, @(x) x, 'x', 1, 2)
%
if isa(f, 'sym') % case of symbolic input
ffun=inline(vectorize(f+0*xvar),char(xvar));
gfun=inline(vectorize(g+0*xvar),char(xvar));
Ffun=inline(vectorize(F+0*xvar),char(xvar),char(yvar));
Gfun=inline(vectorize(G+0*xvar),char(xvar),char(yvar));
oldviewSolid(char(xvar), double(a), double(b), ...
char(yvar), ffun, gfun, char(zvar), Ffun, Gfun)
else
oldviewSolid(char(xvar), double(a), double(b), ...
char(yvar), f, g, char(zvar), F, G)
end
%%%%%%% subfunction goes here %%%%%%
function oldviewSolid(xvar, a, b, yvar, f, g, zvar, F, G)
for counter=0:20
xx = a + (counter/20)*(b-a);
YY = f(xx)*ones(1, 21)+((g(xx)-f(xx))/20)*(0:20);
XX = xx*ones(1, 21);
%% The next lines inserted to make bounding curves thicker.
widthpar=0.5;
if counter==0, widthpar=2; end
if counter==20, widthpar=2; end
```

```
%% Plot curves of constant x on surface patches.
plot3(XX, YY, F(XX, YY).*ones(1,21), 'r', 'LineWidth', widthpar);
hold on
plot3(XX, YY, G(XX, YY).*ones(1,21), 'b', 'LineWidth', widthpar);
end;
%% Now do the same thing in the other direction.
XX = a*ones(1, 21)+((b-a)/20)*(0:20);
%% Normalize sizes of vectors.
YY=0:2; ZZ1=0:20; ZZ2=0:20;
for counter=0:20,
%% The next lines inserted to make bounding curves thicker.
widthpar=0.5;
if counter==0, widthpar=2; end
if counter==20, widthpar=2; end
for i=1:21,
YY(i)=f(XX(i))+(counter/20)*(g(XX(i))-f(XX(i)));
ZZ1(i)=F(XX(i),YY(i));
ZZ2(i)=G(XX(i),YY(i));
end;
plot3(XX, YY, ZZ1, 'r', 'LineWidth',widthpar);
plot3(XX, YY, ZZ2, 'b', 'LineWidth',widthpar);
end;
%% Now plot vertical lines.
for u = 0:0.2:1,
for v = 0:0.2:1,
x=a + (b-a)*u; y = f(a + (b-a)*u) +(g(a + (b-a)*u)-f(a + (b-a)*u))*v;
plot3([x, x], [y, y], [F(x,y), G(x, y)], 'c');
end;
end;
xlabel(xvar)
ylabel(yvar)
zlabel(zvar)
hold off
```

FUNCTION viewSolidone.m:

```
function viewSolidone(zvar, F, G, xvar, f, g, yvar, a, b)
%VIEWSOLID is a version for MATLAB of the routine on page 161
% of "Multivariable Calculus and Mathematica" for viewing the region
% bounded by two surfaces for the purpose of setting up triple integrals.
% The arguments are entered from the inside out.
% There are two forms of the command --- either f, g,
% F, and G can be vectorized functions, or else they can
% be symbolic expressions. xvar, yvar, and zvar can be
% either symbolic variables or strings.
% The variable xvar (x, for example) is on the
% OUTSIDE of the triple integral, and goes between CONSTANT limits a and
b.
% The variable yvar goes in the MIDDLE of the triple integral, and goes
% between limits which must be expressions in one variable [xvar].
% The variable zvar goes in the INSIDE of the triple integral, and goes
% between limits which must be expressions in two
% variables [xvar and yvar]. The lower surface is plotted in red, the
% upper one in blue, and the "hatching" in cyan.
%
% Examples: viewSolid(z, 0, (x+y)/4, y, x/2, x, x, 1, 2)
% gives the picture on page 163 of "Multivariable Calculus and
Mathematica"
% and the picture on page 164 of "Multivariable Calculus and Mathematica"
% can be produced by
% viewSolid(z, x^2+3*y^2, 4-y^2, y, -sqrt(4-x^2)/2, sqrt(4-x^2)/2, ...
% x, -2, 2,)
% One can also type viewSolid('z', @(x,y) 0, ...
% @(x,y)(x+y)/4, 'y', @(x) x/2, @(x) x, 'x', 1, 2)
%
if isa(f, 'sym') % case of symbolic input
ffun=inline(vectorize(f+0*yvar),char(yvar));
gfun=inline(vectorize(g+0*yvar),char(yvar));
Ffun=inline(vectorize(F+0*xvar),char(xvar),char(yvar));
Gfun=inline(vectorize(G+0*xvar),char(xvar),char(yvar));
oldviewSolid(char(yvar),double(a), double(b), ...
char(xvar), ffun, gfun, char(zvar), Ffun, Gfun)
else
oldviewSolid(char(yvar),double(a),double(b),char(xvar), f, g, char(zvar),
F, G)
end
%%%%%%% subfunction goes here %%%%%%
function oldviewSolid(yvar,a , b, xvar, f, g, zvar, F, G)
for counter=0:30
yy= a + (counter/30)*(b-a);
XX = f(yy)*ones(1, 31)+((g(yy)-f(yy))/30)*(0:30);
YY = yy*ones(1, 31);
%% The next lines inserted to make bounding curves thicker.
widthpar=0.5;
```

```
if counter==0, widthpar=2; end
if counter==20, widthpar=2; end
%% Plot curves of constant x on surface patches.
plot3(YY,XX, F(XX, YY).*ones(1,31), 'r', 'LineWidth', widthpar);
hold on
plot3(YY,XX, G(XX, YY).*ones(1,31), 'b', 'LineWidth', widthpar);
end;
%% Now do the same thing in the other direction.
YY = a*ones(1, 31)+((b-a)/30)*(0:30);
%% Normalize sizes of vectors.
XX=0:2; ZZ1=0:30; ZZ2=0:30;
for counter=0:30,
%% The next lines inserted to make bounding curves thicker.
widthpar=0.5;
if counter==0, widthpar=2; end
if counter==30, widthpar=2; end
for i=1:31,
XX(i)=f(YY(i))+(counter/30)*(g(YY(i))-f(YY(i)));
ZZ1(i)=F(YY(i),XX(i));
ZZ2(i)=G(YY(i),XX(i));
end;
plot3(YY,XX, ZZ1, 'r', 'LineWidth',widthpar);
plot3(YY,XX, ZZ2, 'g', 'LineWidth',widthpar);
end;
%% Now plot vertical lines.
for u = 0:0.09:1,
for v = 0:0.09:1,
y=a + (b-a)*u; x = f(a + (b-a)*u) +(g(a + (b-a)*u)-f(a + (b-a)*u))*v;
plot3([y, y], [x, x], [F(x,y), G(x, y)], 'c');
end;
end;
xlabel(xvar)
ylabel(yvar)
zlabel(zvar)
hold off
```

EXAMPLE PROBLEMS

Question 1:

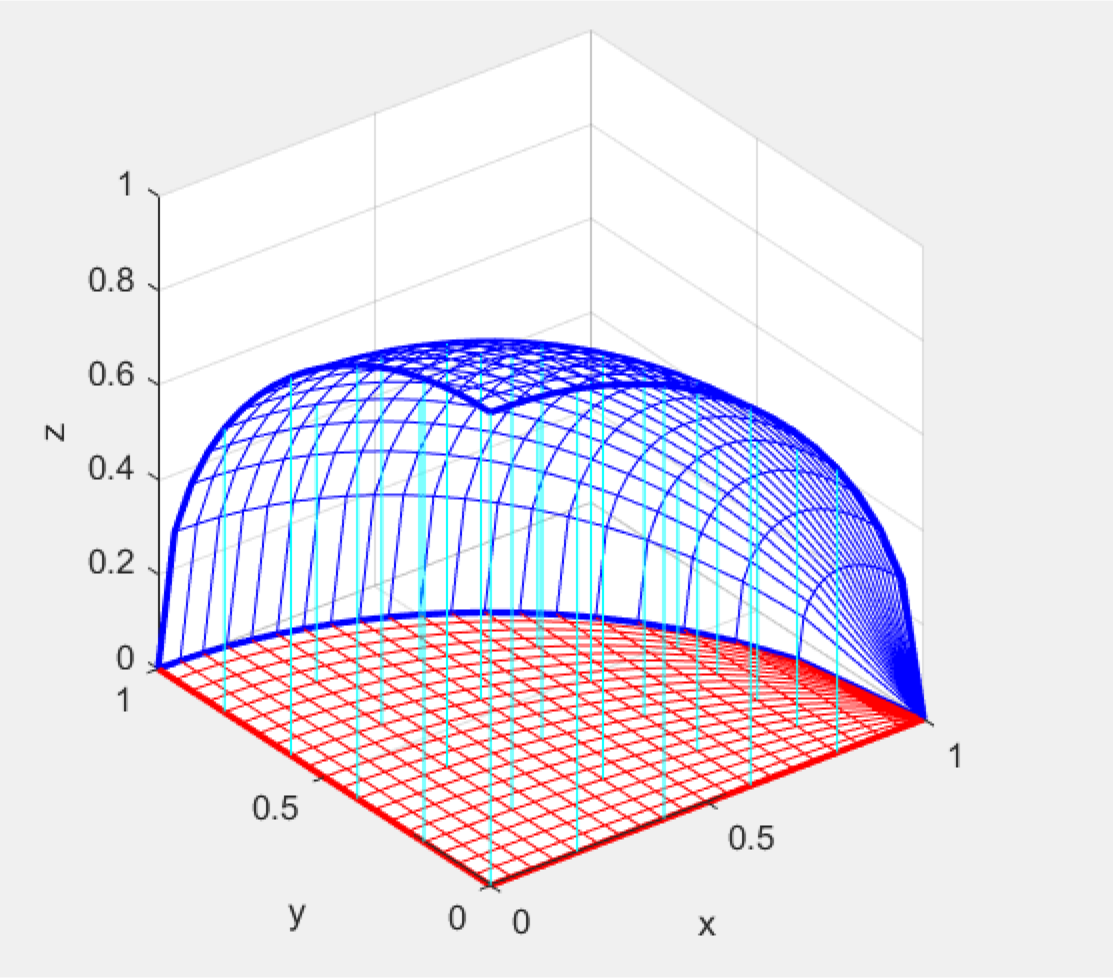
MATLAB CODE

```
syms x y z
vol=8*int(int(sqrt(1-x^2-y^2),y,0,sqrt(1-x^2)),x,0,1)
viewSolid(z,0+0*x*y,sqrt(1-x^2-y^2),y,0+0*x,sqrt(1-x^2),x,0,1);
axis equal;
grid on;
```

Output

vol =

(4*pi)/3



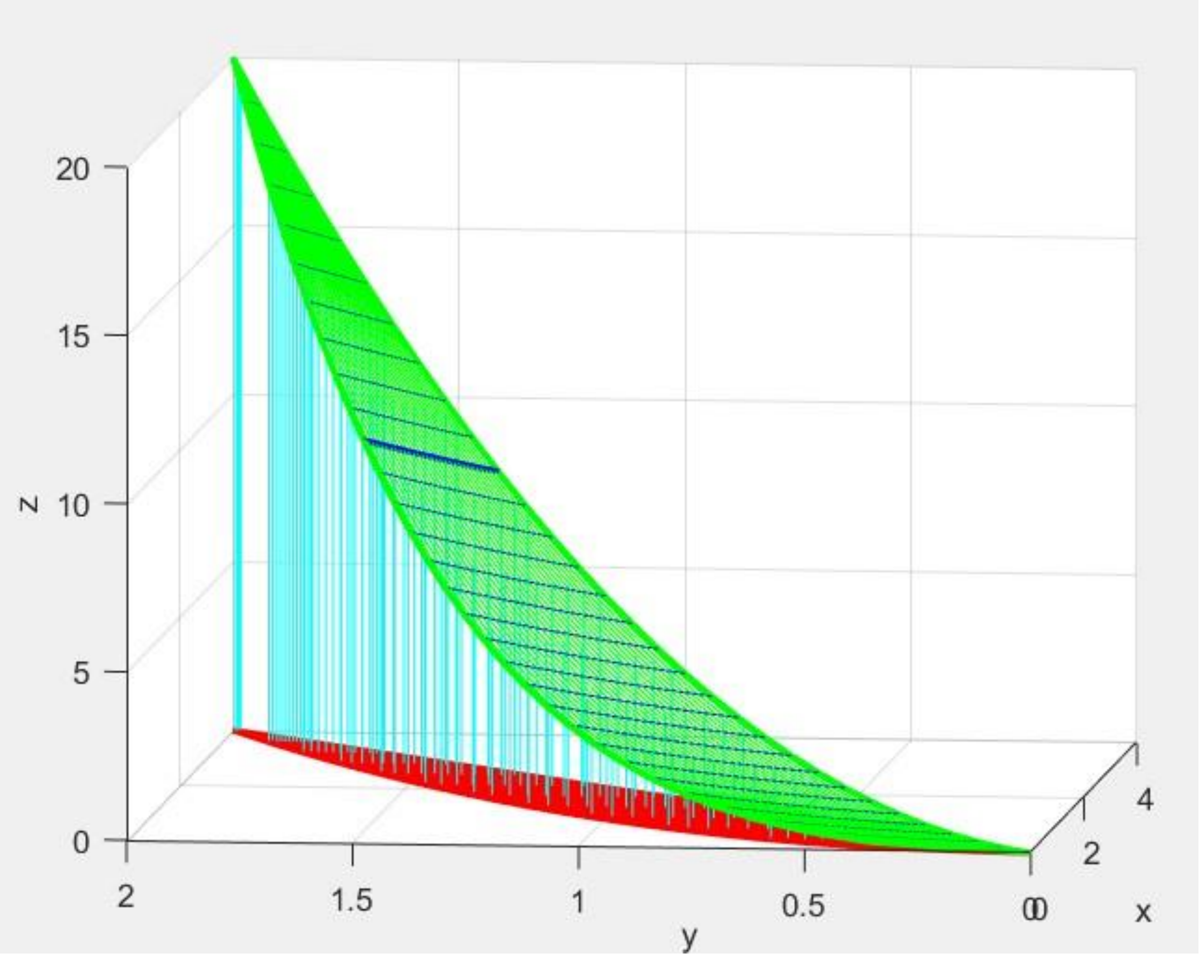
Question 2:

MATLAB CODE

```
clc
clear all
format compact
syms x y z
vol = int(int(x^2+y^2, x,y/2,sqrt(y)), y, 0, 4)
viewSolidone(z,0+0*x*y,x^2+y^2,x,y/2,sqrt(y),y,0,4);
grid on;
```

Output

vol =
216/35



Question 3:

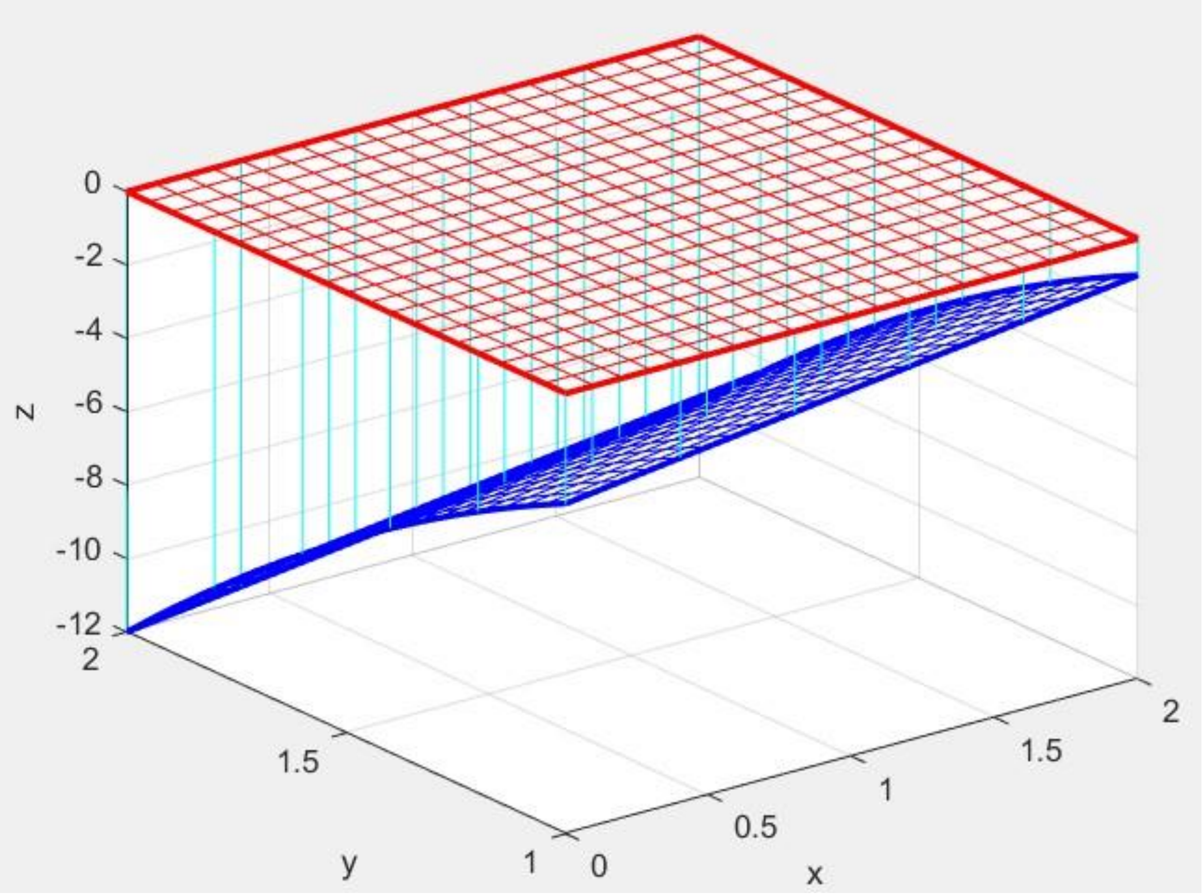
MATLAB CODE

```
clc
clear all
format compact
syms x y z
vol = int(int(x-3*y^2, x,0,2), y,1,2)
viewSolid(z,0+0*x+0*y,x-3*y^2+0*y,y,1+0*x,2+0*x,x,0,2)
grid on
```

Output

vol =

-12



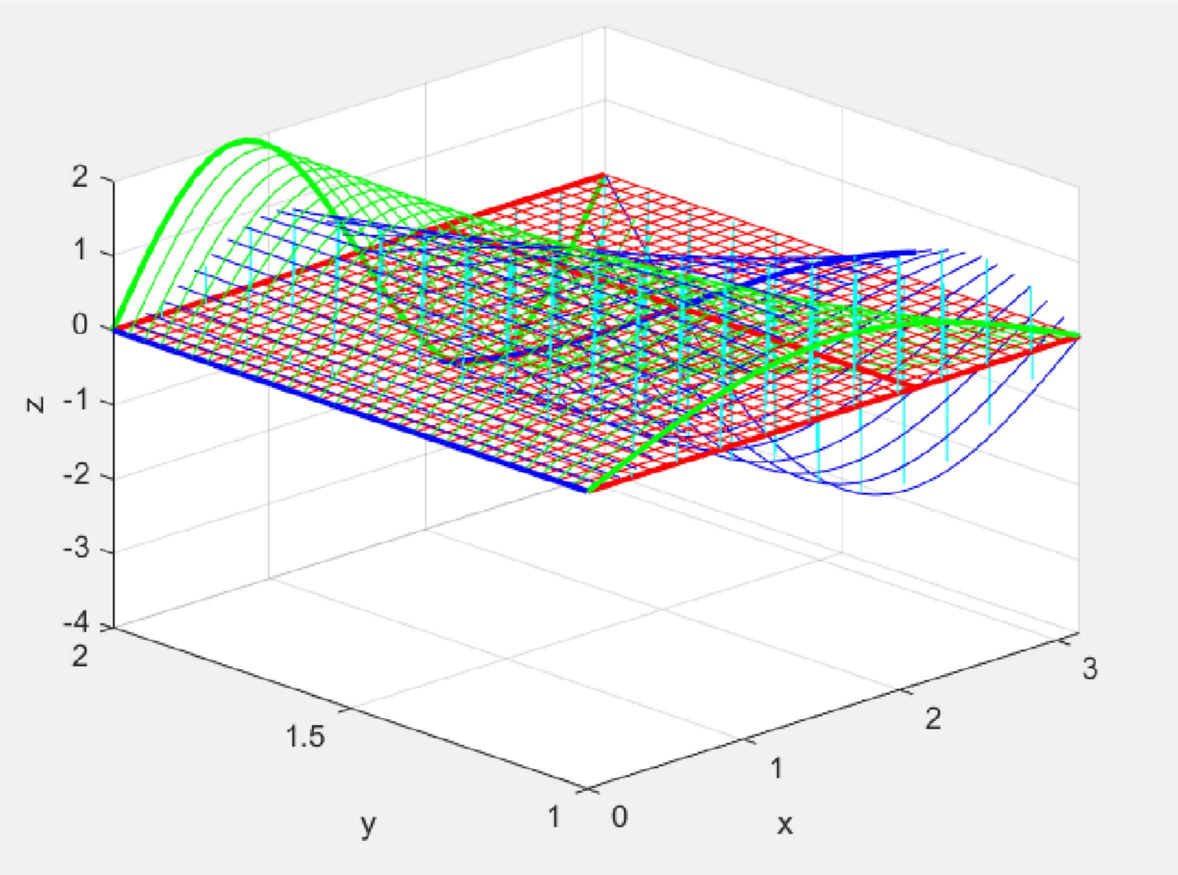
Question 4:
MATLAB CODE

```
clc
clear all
format compact
syms x y z
vol =int(int(y*sin(x*y),x,1,2),y,0,pi)
viewSolidone(z,0*x+0*y,y*sin(x*y),x,1+0*y,2+0*y,y,0,pi)
grid on
```

Output

vol =

0



Question 5:

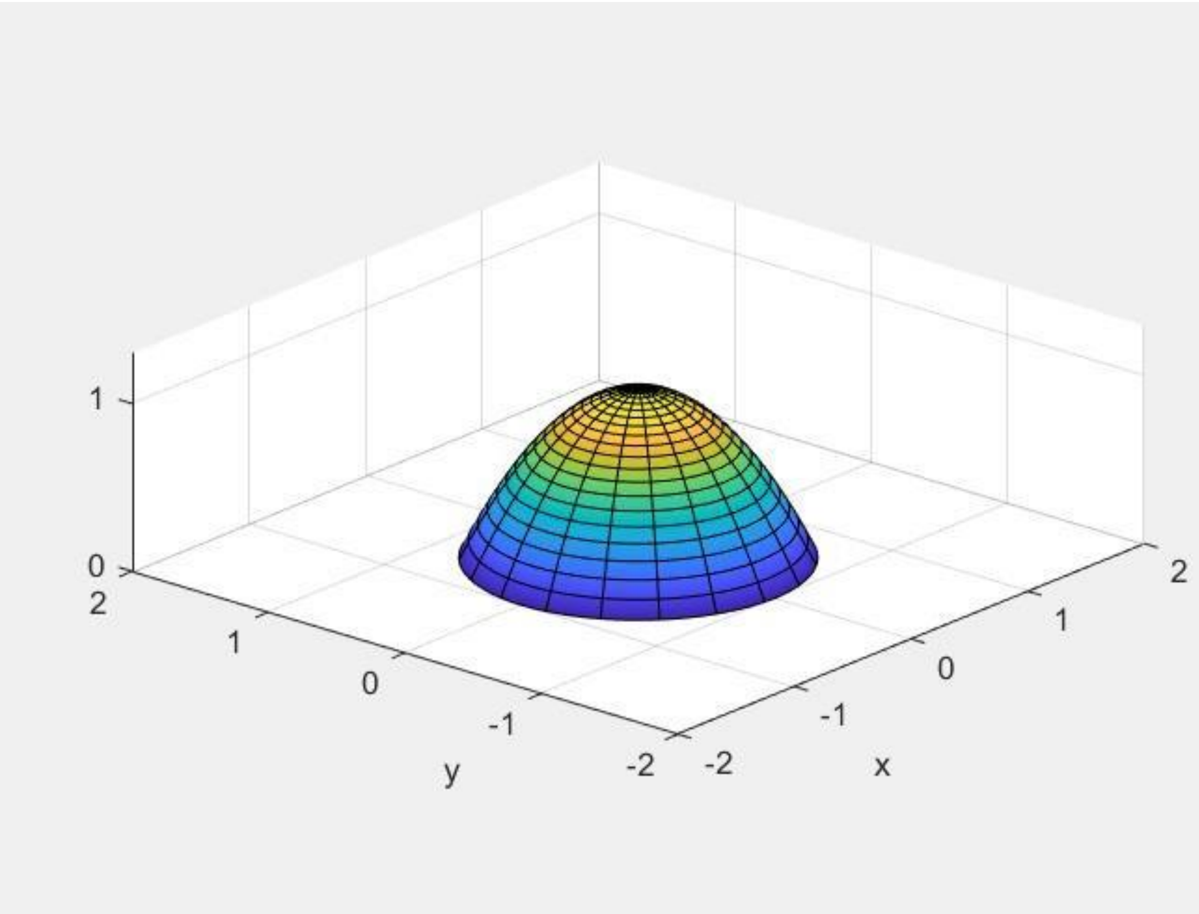
MATLAB CODE

```
clc
clear all
format compact
syms r theta
V=int(int((1-r^2)*r,r,0,1),theta,0,2*pi)
fsurf(r*cos(theta),r*sin(theta),1-r^2,[0 1 0 2*pi],'MeshDensity',20)
axis equal; axis([-2 2 -2 2 0 1.3])
xticks(-2:2); yticks(-2:2); zticks(0:1.3)
xlabel('x'); ylabel('y')
```

Output

V =

pi/2



Question 6:

MATLAB CODE

```
clc
clear all
format compact
syms r theta z r1
V=int(int((r^2)*r,r,0,2*cos(theta)),theta,-pi/2,pi/2)
r=2*cos(theta),x=r*cos(theta),y=r*sin(theta)
fsurf(x,y,z,[0 2*pi 0 1],'MeshDensity',16)
axis equal;xlabel('x'); ylabel('y'); zlabel('z')
zticks(0:1.5)
hold on
fsurf(r1*cos(theta),r1*sin(theta),r1^2,[0 1 0 2*pi], 'MeshDensity',20)
```

Output

V =

(3*pi)/2

r =

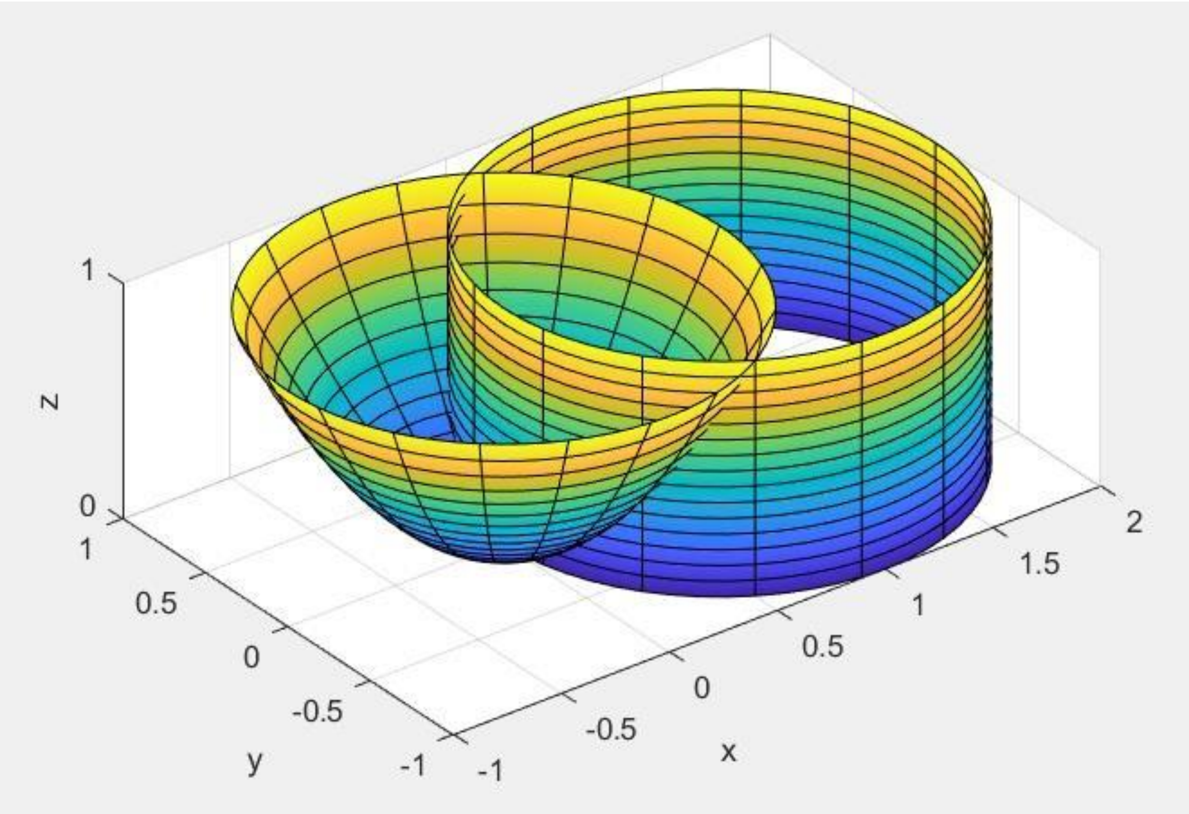
2*cos(theta)

x =

2*cos(theta)^2

y =

2*cos(theta)*sin(theta)



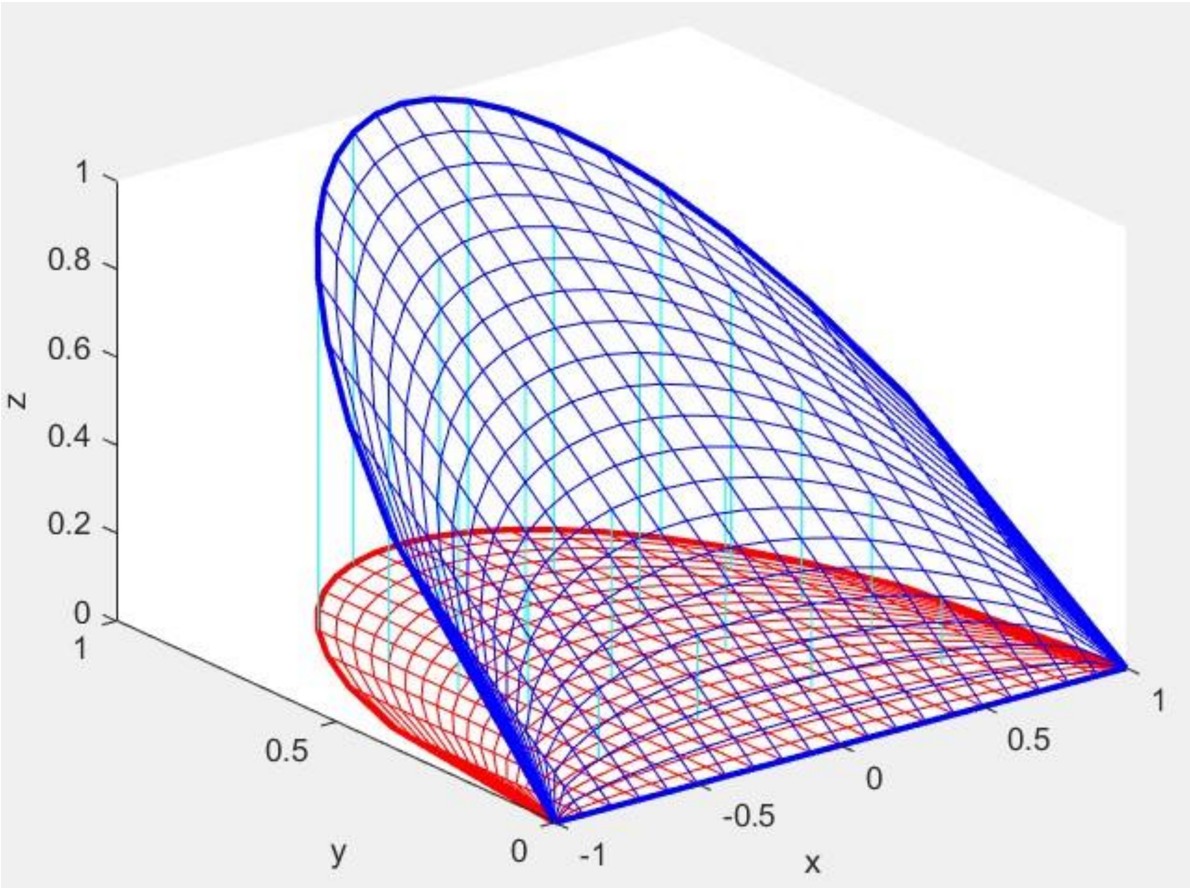
EXERCISE PROBLEMS
Question 1:
MATLAB CODE

```
clc
clear all
syms x y z
I = int(int(y,y,0,sqrt(1-x^2)),x,-1,1)
viewSolid(z,0+0*x+0*y,y,y,0+0*x,sqrt(1-x^2),x,-1,1)
```

Output

I =

2/3



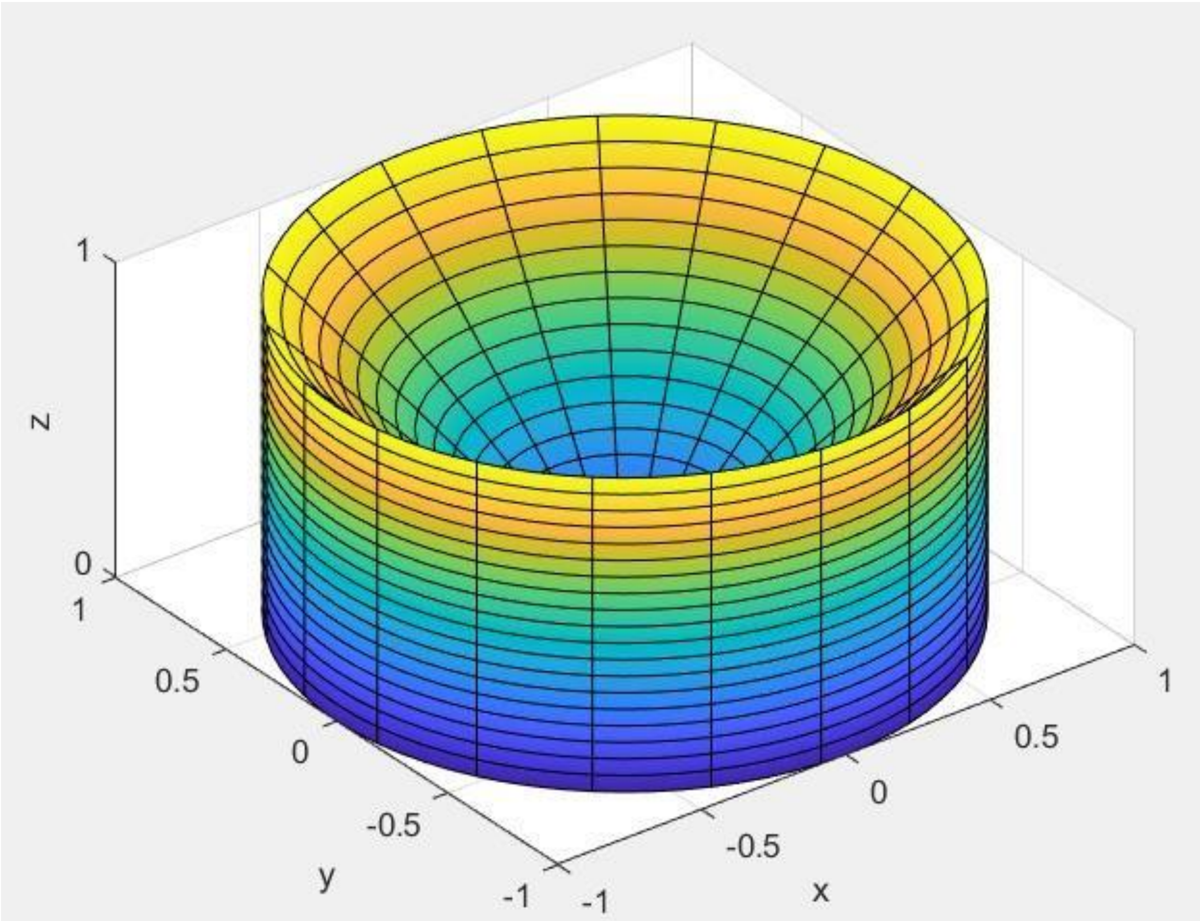
Question 2:

MATLAB CODE

```
clc
clear all
format compact
syms r theta z r1
V = int(int((r^2), r, 0, 1 ), theta, 0, 2*pi)
fsurf(r*cos(theta),r*sin(theta),r, [0 1 0 2*pi], 'MeshDensity', 20)
axis equal
xlabel('x')
ylabel('y')
zlabel('z')
zticks(0:5)
hold on
r = 1
x = r*cos(theta)
y = r*sin(theta)
fsurf(x,y,z, [0 2*pi 0 1], 'MeshDensity', 20)
```

Output

```
V =
(2*pi)/3
r =
   1
x =
cos(theta)
y =
sin(theta)
```



1
z
0
1
0.5
0
-0.5
y
-1
-1
-0.5
0
0.5
1
x

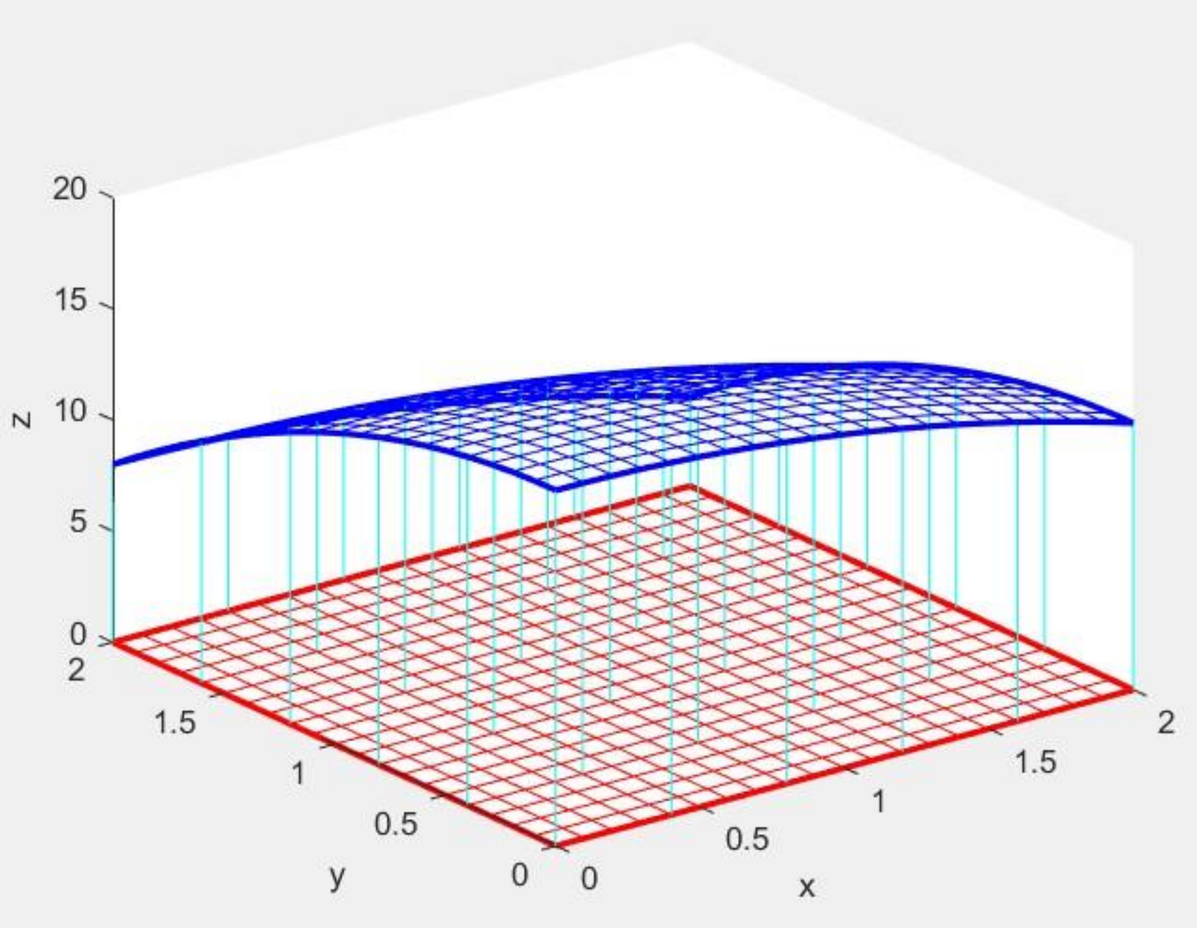
Question 3:
MATLAB CODE

```
clc
clear all
format compact
syms x y z
int(int(16-x^2-2*y^2,x,0,2),y,0,2)
viewSolid(z, 0+0*x+0*y, 16-x^2-2*y^2,y,0+0*x,2+0*x,x,0,2)
```

Output

ans =

48



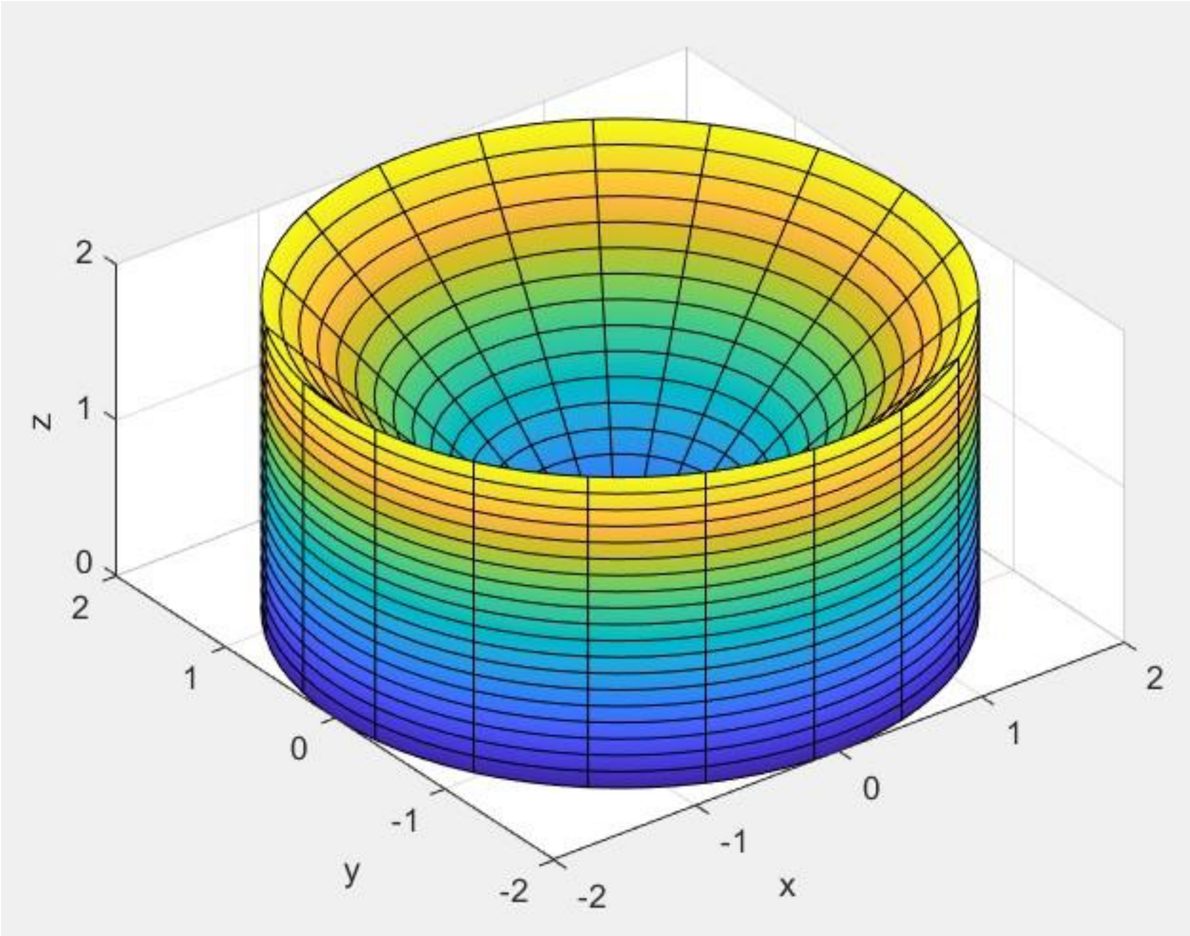
Question 4:

MATLAB CODE

```
clc
clear all
format compact
syms r theta z r1
vol = int(int((r^2), r, 0, 2 ), theta, 0, 2*pi)
fsurf(r*cos(theta),r*sin(theta),r, [0 2 0 2*pi], 'MeshDensity', 20)
axis equal
xlabel('x')
ylabel('y')
zlabel('z')
zticks(0:5)
hold on
r = 2
x = r*cos(theta)
y = r*sin(theta)
fsurf(x,y,z, [0 2*pi 0 2], 'MeshDensity', 20)
```

Output

```
vol =
(16*pi)/3
r =
   2
x =
2*cos(theta)
y =
2*sin(theta)
```



z
2
1
0
y
2
1
0
-1
-2
x
-2
-1
0
1
2