Experiment 9

Vector Fields and Their Integration

I. Aim

To plot 2-D and 3-D Vector Fields by making function files and Calculating work done by vector field.

II. MATLAB Function Code for 2-D Vector Field Plot:

```
function h = vect2D(f,xL,yL)
    P = inline(vectorize(f(1)),'x','y');
    Q = inline(vectorize(f(2)),'x','y');
    [X,Y] = meshgrid(linspace(xL(1),xL(2),8),(linspace(yL(1),yL(2),8)));
    U = P(X,Y);
    V = Q(X,Y);
    h = quiver(X,Y,U,V,1);
end
```

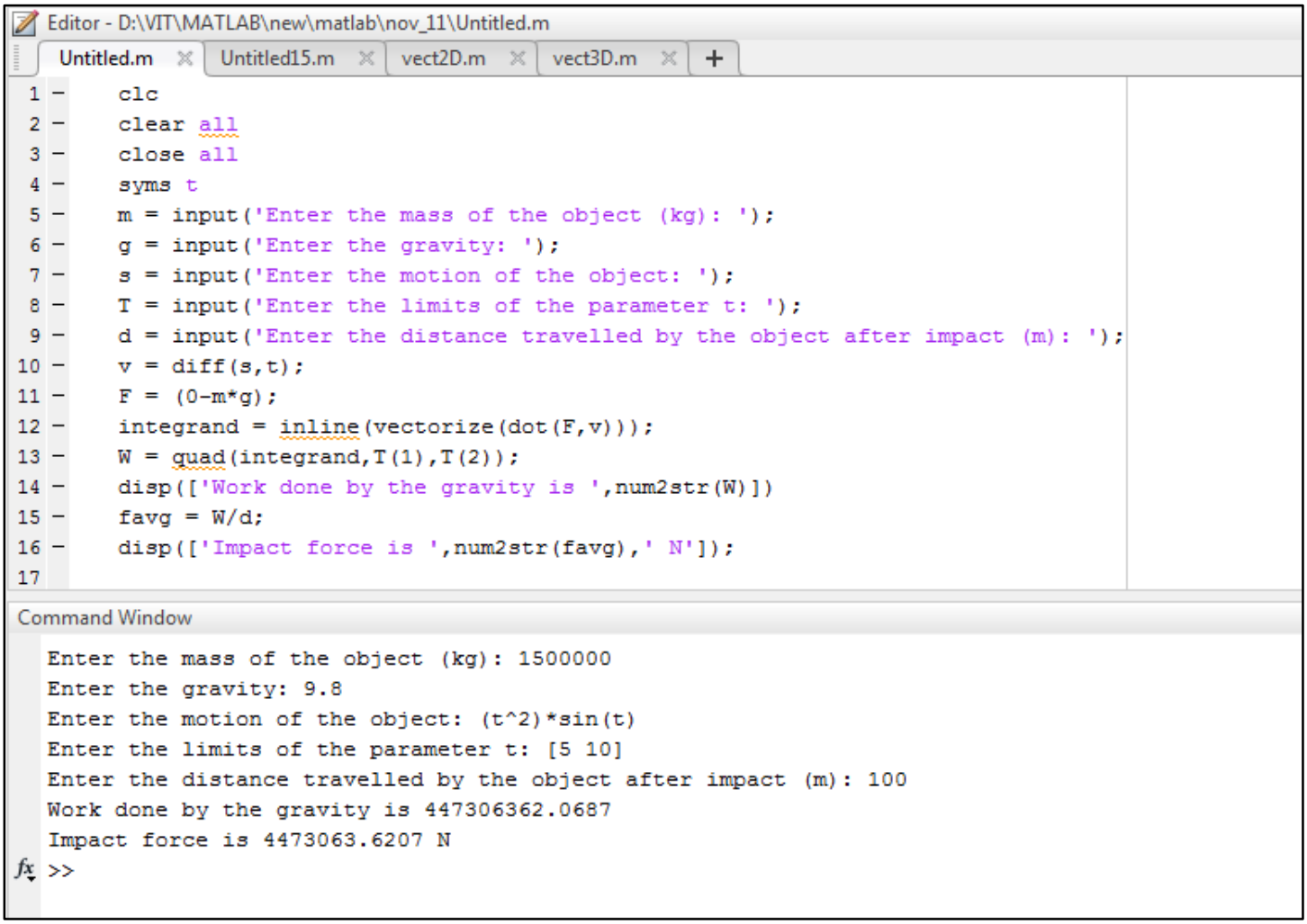
III. MATLAB Function Code for 3-D Vector Field Plot:

```
function h = vect3D(f,xL,yL,zL)
    P = inline(vectorize(f(1)),'x','y','z');
    Q = inline(vectorize(f(2)),'x','y','z');
    R = inline(vectorize(f(3)),'x','y','z');
    [X,Y,Z] =
meshgrid(linspace(xL(1),xL(2),8),(linspace(yL(1),yL(2),8)),linspace(zL(1),zL(2),8));
    U = P(X,Y,Z);
    V = Q(X,Y,Z);
    W = R(X,Y,Z);
    h = quiver3(X,Y,Z,U,V,W,1);
end
```

IV. MATLAB code for finding impact force along with work done by gravity

```
clc
clear all
close all
syms t
m = input('Enter the mass of the object (kg): ');
g = input('Enter the gravity: ');
s = input('Enter the motion of the object: ');
T = input('Enter the limits of the parameter t: ');
d = input('Enter the distance travelled by the object after impact (m): ');
v = diff(s,t);
F = [0 -m*g];
integrand = inline(vectorize(dot(F,v)));
```

```
W = quad(integrand,T(1),T(2));
disp(['Work done by the gravity is',num2str(W)])
favg = W/d;
disp(['Impact force is',num2str(favg),'N']);
```



V. MATLAB code for calculating work done by vector field

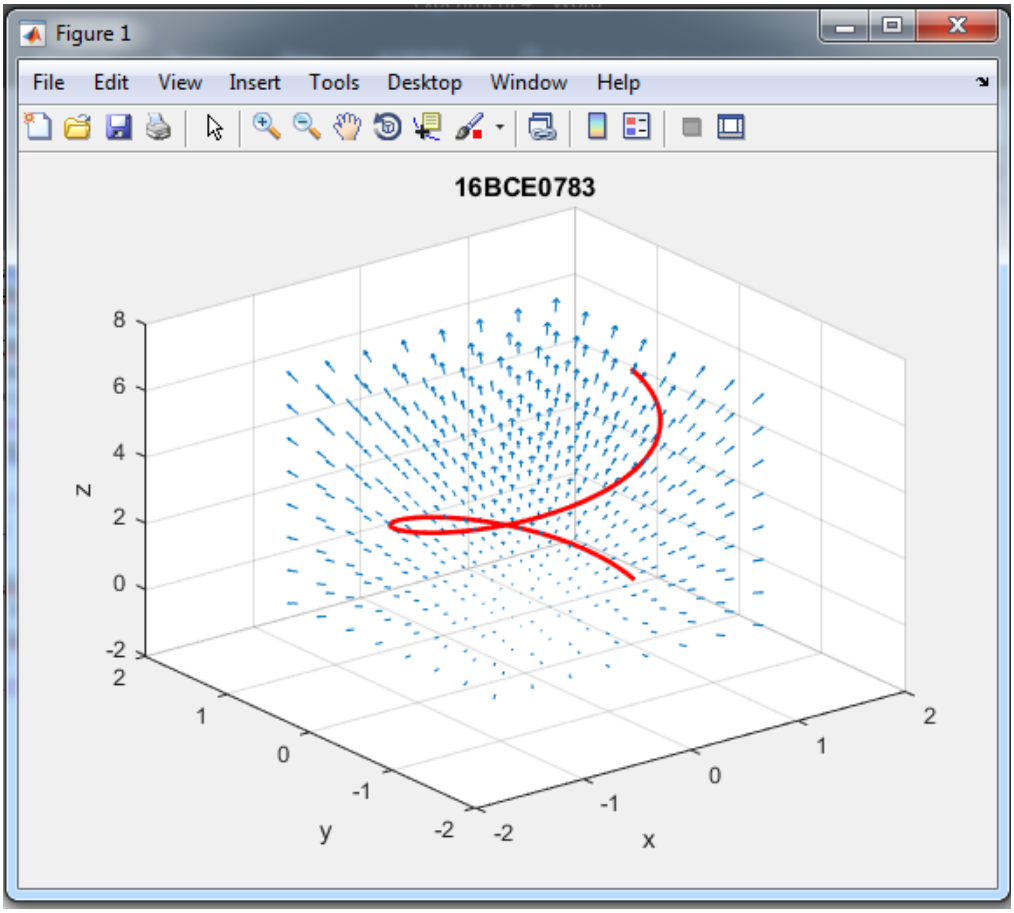
```
clc
clear all
close all
syms x y z t
F = input('Enter the field components in cartesian coordinates: ');
C = input('Enter the parametric form of C(t): ');
L = input('Limits of parameter t: ');
T = linspace(L(1),L(2),101);
X = eval(subs(C(1),t,T)); %x-component of C(t)
Y = eval(subs(C(2),t,T));  %y-component of C(t)
xL = [min(X)-0.2,max(X)+0.2]; %Setting limits for plotting purpose
yL = [min(Y)-0.2,max(Y)+0.2];   %Setting limits for plotting purpose
if numel(F) == 2
    FF = subs(F,{x,y},C); % Calculating F(C(t)) for line integral
    vect2D(F,xL,yL);  % Calling function
    hold on;
```

```
    plot(X,Y,'-r','LineWidth',2); %plotting the curve c
elseif numel(F) == 3
    FF = subs(F,{x,y,z},C);
    Z = eval(subs(C(3),t,T));  %z-component of C(t)
    zL = [min(Z)-0.2,max(Z)+0.2];
    vect3D(F,xL,yL,zL);
    hold on;
    plot3(X,Y,Z,'-r','LineWidth',2);
else
    disp(['Function entered is not a 2D or 3D vector field.'])
    return
end
axis on; xlabel('x');ylabel('y');zlabel('z')
intgrnd = inline(vectorize(dot(FF,diff(C,t))),'t');
WD = quad(intgrnd,L(1),L(2));
disp(['The work done by vector field is ',num2str(WD)]);
title('16BCE0783')
```

```
14 -        FF = subs(F,{x,y},C); % Calculating F(C(t)) for line integral
15 -        vect2D(F,xL,yL);  % Calling function
16 -        hold on;
17 -        plot(X,Y,'-r','LineWidth',2); %plotting the curve c
```

Command Window

```
Enter the field components in cartesian coordinates: [x y z]
Enter the parametric form of C(t): [cos(t) sin(t) t]
Limits of parameter t: [0 2*pi]
The work done by vector field is 19.7392
fx >>
```



VI. Question - Answers:

Q1 Answer

```
Command Window
>> %converting into parametric form and let a = 1
>> %x = cos(t)^3
>> %y = sin(t)^3
>> %16BCE0783
>> a1 = [cos(t)^3,sin(t)^3]
Undefined function or variable 't'.

>> syms t
>> a1 = [cos(t)^3,sin(t)^3]

a1 =

[ cos(t)^3, sin(t)^3]

>> area = int(a1(1)*diff(a1(2)),t,0,2*pi)

area =

(3*pi)/8

fx >>
```

Q2 Answer

By using the code in section V

```
33 -     title('16BCE0783')
34       %will not work for constant vector fields
Command Window
Enter the field components in cartesian coordinates: [-16*y + sin(x^2) 4*exp(y)+3*x^2]
Enter the parametric form of C(t): [cos(t)/sqrt(2) sin(t)/sqrt(2)]
Limits of parameter t: [-pi/4 pi/4]
The work done by vector field is 7.7019
fx >>
```

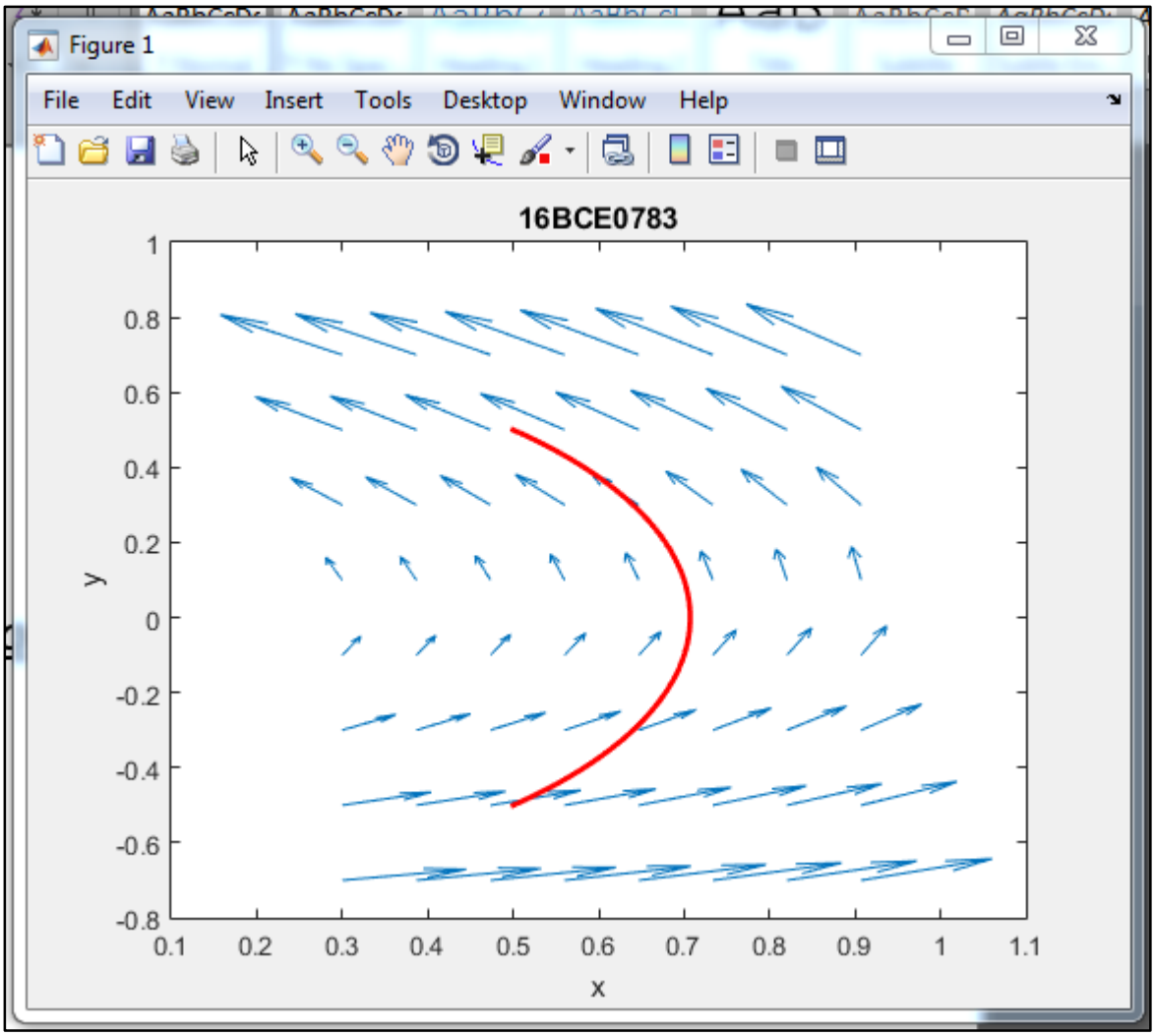


Figure 1
File Edit View Insert Tools Desktop Window Help
16BCE0783
y
x
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 1.1