```
%Calvin Ankener and Gino Rospigliosi
%Math 246
%Section 0423
%TA: Thien Ngo
%MATLAB Project B
```

```
%Problem 3:
```

```
%Part a:
soll_3 = dsolve('Dy = (2*t - y)/t', 'y(1) = c', 't')
```

```
soll_3 =

t + (c - 1)/t
```

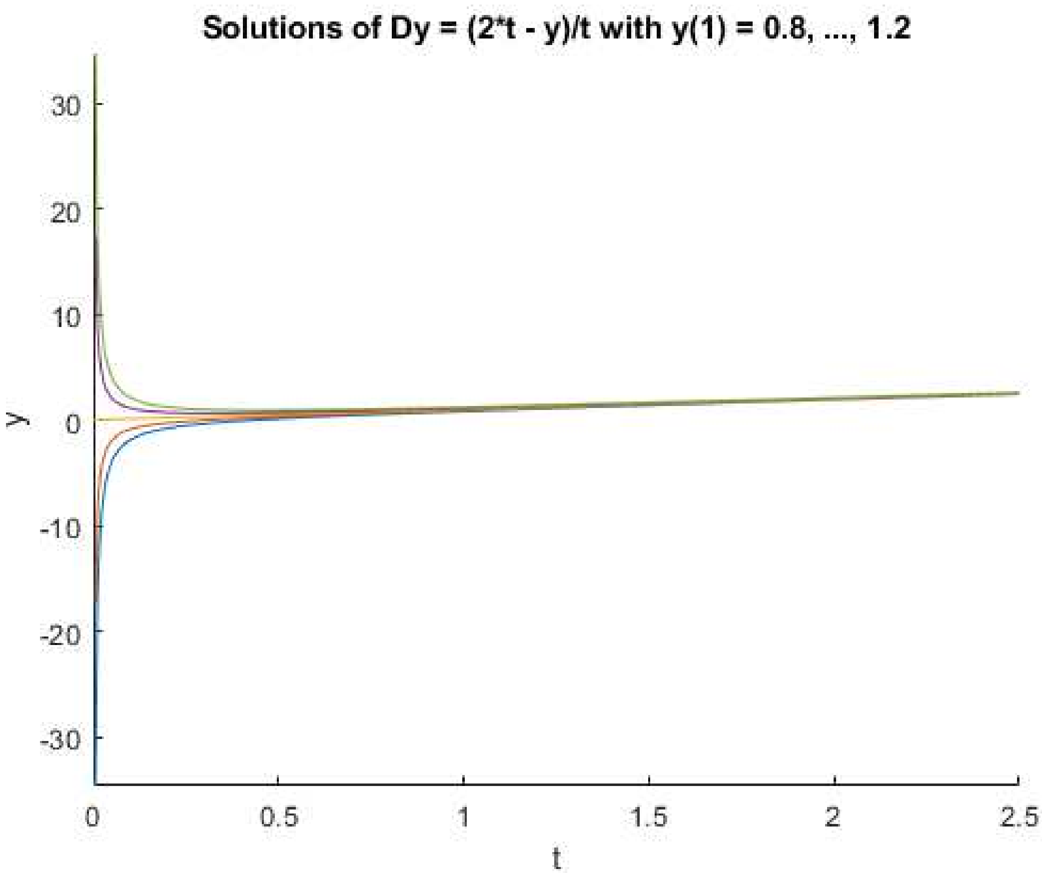
```
%Part b:
syms t;
syms c;
%Here I evaluate the solution when c = 0.8 and t = 0.01.
subs(soll_3, [c, t], [.8, .01])
%Here I evaluate the solution when c = 0.8 and t = 0.1.
subs(soll_3, [c, t], [.8, .1])
%Here I evaluate the solution when c = 0.8 and t = 1.
subs(soll_3, [c, t], [.8, 1])
%Here I evaluate the solution when c = 0.8 and t = 10.
subs(soll_3, [c, t], [.8, 10])
%Here I evaluate the solution when c = 1 and t = 0.01.
subs(soll_3, [c, t], [1, .01])
%Here I evaluate the solution when c = 1 and t = 0.1.
subs(soll_3, [c, t], [1, .1])
%Here I evaluate the solution when c = 1 and t = 1.
subs(soll_3, [c, t], [1, 1])
%Here I evaluate the solution when c = 1 and t = 10.
subs(soll_3, [c, t], [1, 10])
%Here I evaluate the solution when c = 1.2 and t = 0.01.
subs(soll_3, [c, t], [1.2, .01])
%Here I evaluate the solution when c = 1.2 and t = 0.1.
subs(soll_3, [c, t], [1.2, .1])
%Here I evaluate the solution when c = 1.2 and t = 1.
subs(soll_3, [c, t], [1.2, 1])
%Here I evaluate the solution when c = 1.2 and t = 10.
subs(soll_3, [c, t], [1.2, 10])
```

```
ans =
```

```
-1999/100

ans =

-19/10

ans =

4/5

ans =

499/50

ans =

1/100

ans =

1/10

ans =

1

ans =

10

ans =

2001/100

ans =

21/10

ans =

6/5

ans =

501/50
```

Part c:

```
hold on;
syms t;
for cval = .8:.1:1.2
    ezplot(subs(soll_3, 'c', cval), [0 2.5])
end
axis tight;
title('Solutions of Dy = (2*t - y)/t with y(1) = 0.8, ..., 1.2');
xlabel('t');
ylabel('y');
hold off;
```




```
%Part d:
%As t approaches infinity, the solutions approach zero. As t approaches 0
%from the positive right-hand side, the solutions with c = 1.1 and 1.2
%approach infinity, the solution with c = 1.0 approaches 0, and the
%solutions with c = 0.8 and 0.9 approach negative infinity.
```

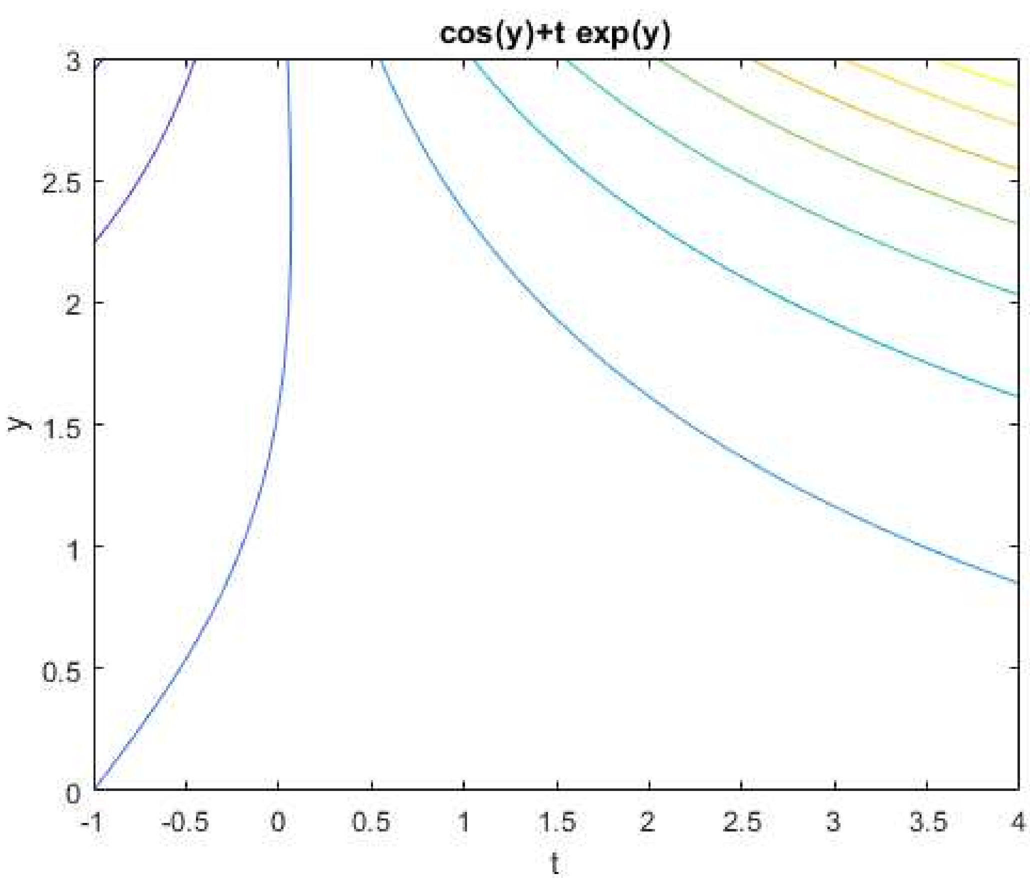
```
%Problem 7:
```

```
%Part a:
soll_7=dsolve('Dy = (-exp(y))/(t*exp(y)-sin(y))','t')
```

```
Warning: Unable to find explicit solution. Returning implicit solution instead.

soll_7 =

solve(cos(y) + t*exp(y) == C11, y)
```

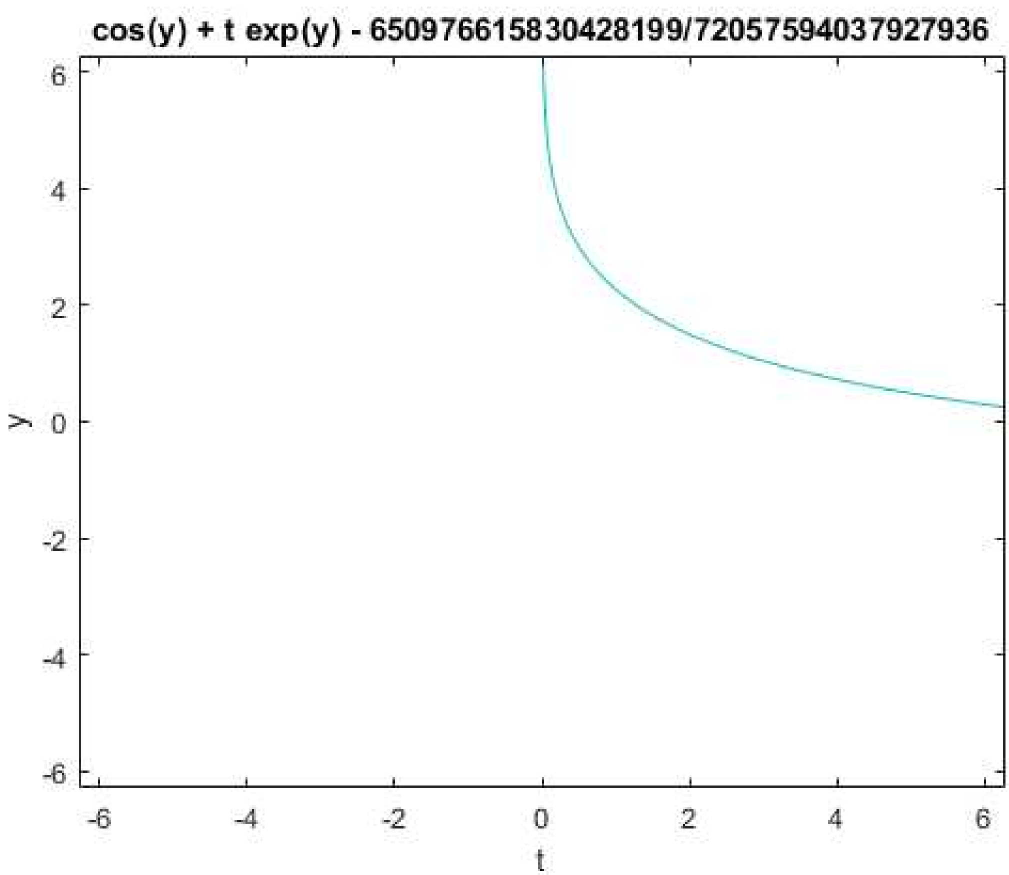
```
%Part b:
fun=@(t,y)cos(y) + t*exp(y)
ezcontour(fun,[-1,4],[0,3])
```

```
fun =

  function_handle with value:

    @(t,y)cos(y)+t*exp(y)

Warning: Function failed to evaluate on array inputs; vectorizing the function
may speed up its evaluation and avoid the need to loop over array elements.
```

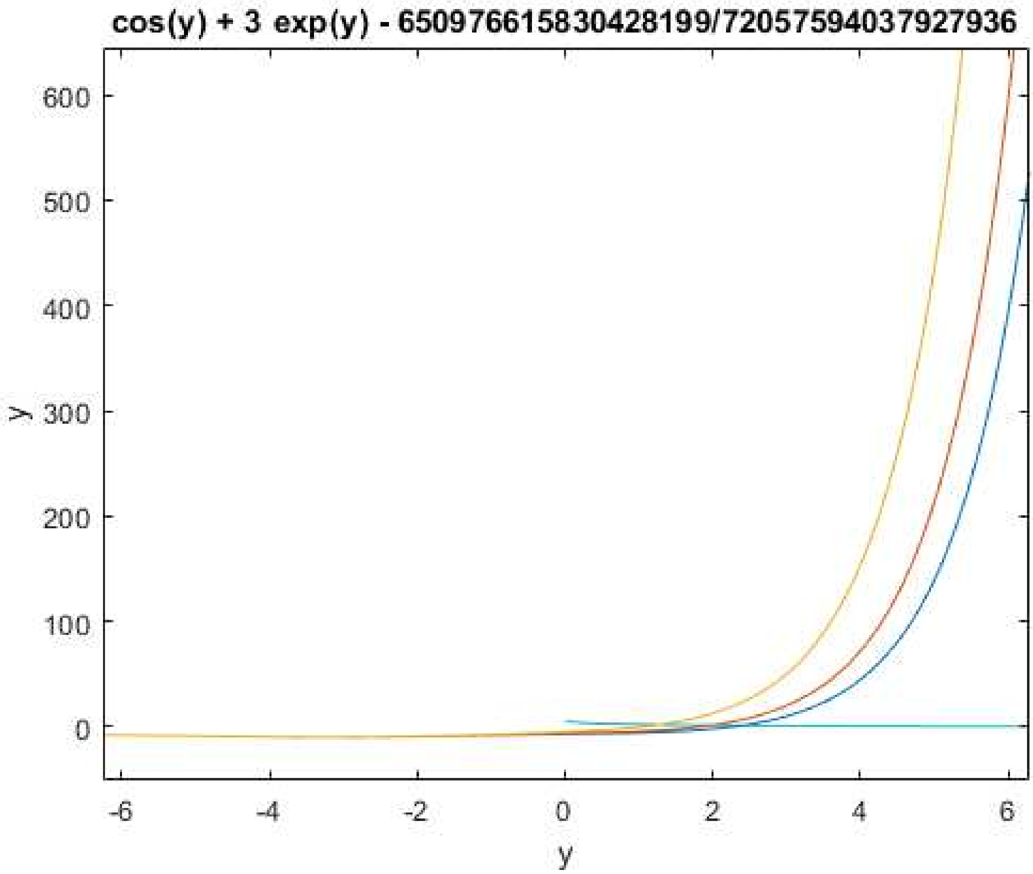


```
%Part c:
syms y t;
soll_7c=dsolve('Dy = (-exp(y))/(t*exp(y)-sin(y))','y(2)=1.5','t')
ezplot(cos(y) + t*exp(y)- cos(3/2) - 2*exp(3/2))
```

```
Warning: Unable to find explicit solution. Returning implicit solution instead.

soll_7c =

solve(cos(y) + t*exp(y) == cos(3/2) + 2*exp(3/2), y)
```

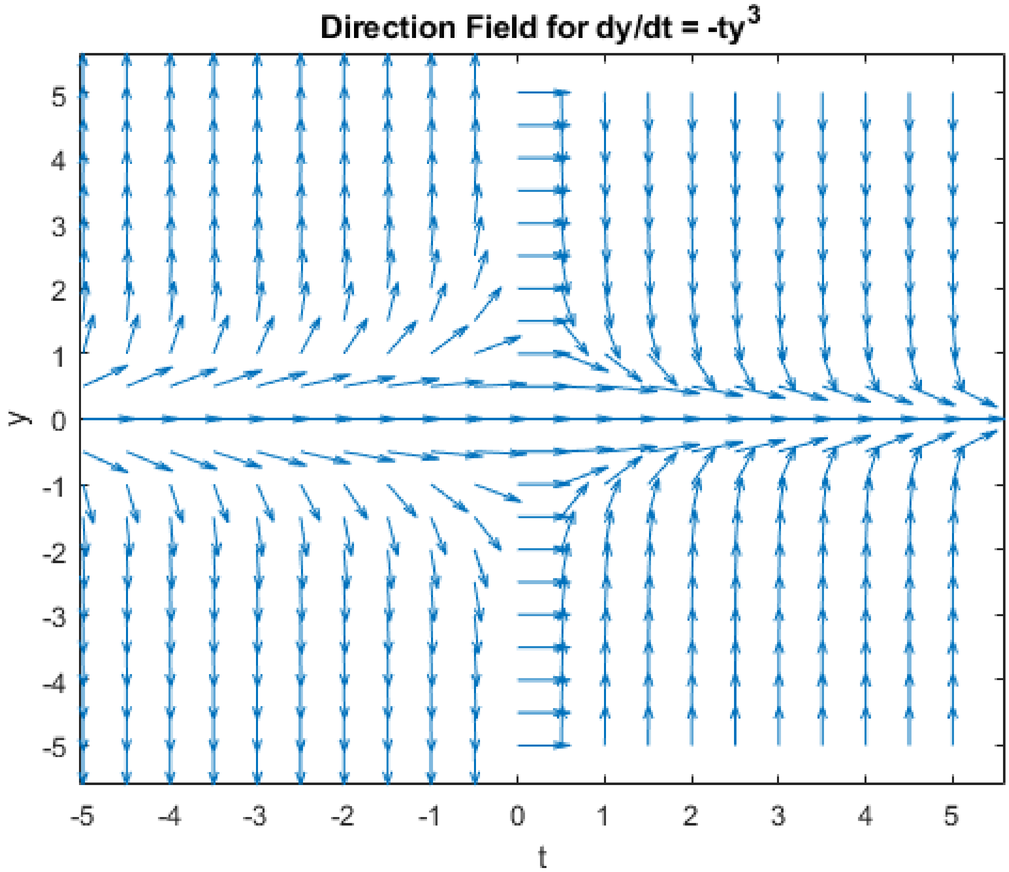



```
%Part d:
syms y t;
ezplot(cos(y) + t*exp(y)- cos(3/2) - 2*exp(3/2))
hold on
ezplot(cos(y) + 1*exp(y)- cos(3/2) - 2*exp(3/2))
ezplot(cos(y) + 1.5*exp(y)- cos(3/2) - 2*exp(3/2))
ezplot(cos(y) + 3*exp(y)- cos(3/2) - 2*exp(3/2))
hold off
```




```
%Problem 16:
```

```
%Part a:
[t,y] = meshgrid(-5:0.5:5,-5:0.5:5);
dy = -t.*y.^3;
dt = ones(size(dy));
dyu = dy./sqrt(dt.^2+dy.^2);
dtu = dt./sqrt(dt.^2+dy.^2);
quiver(t,y,dtu,dyu)
axis tight;
xlabel('t');
ylabel('y');
title('Direction Field for dy/dt = -ty^3');
%There is a constant solution y(0)=0. If y(0) > 0, the solution approaches
%zero as t increases. If y(0) < 0, the solution approaches zero as t
%increases.
```




```
%Part b:
%The solutions follow the curve of the direction field.
```

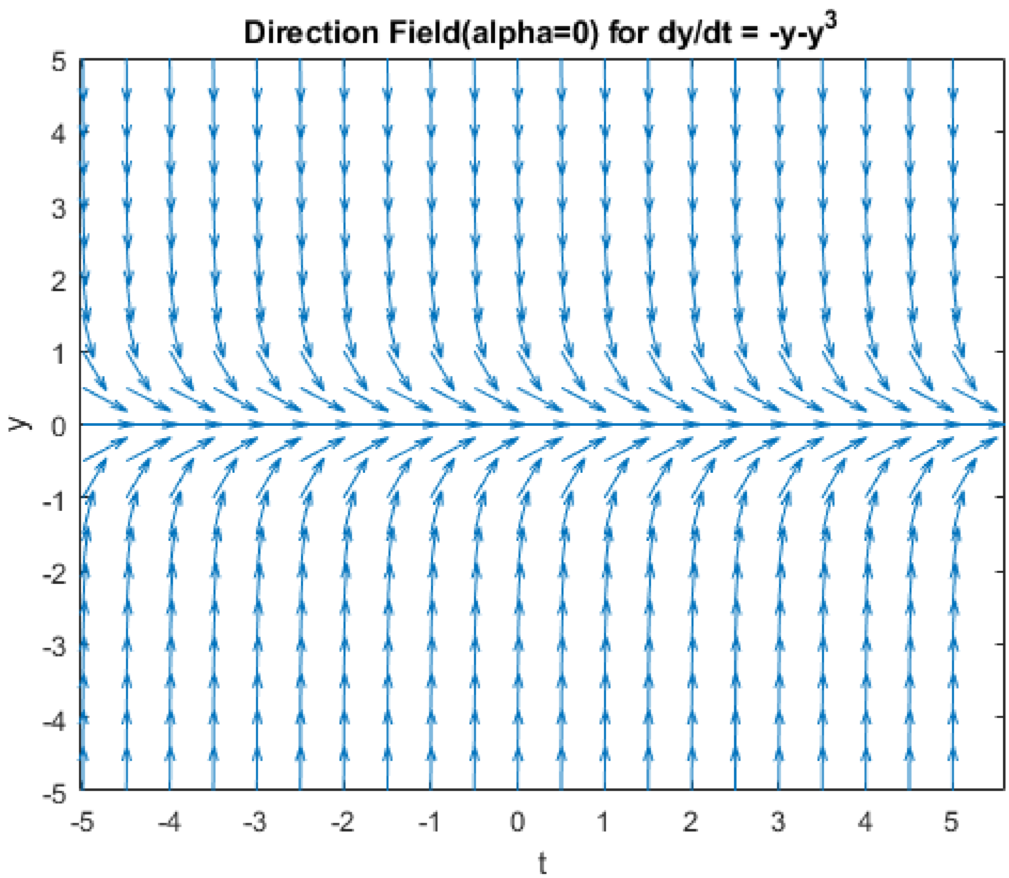
```
%Part c:
soll_16 = dsolve('Dy = -t*y^3', 'y(0) = 1/sqrt(c)', 't')
```

```
soll_16 =

  (2^(1/2)*(1/(t^2/2 + c/2))^(1/2))/2
 -(2^(1/2)*(1/(t^2/2 + c/2))^(1/2))/2
```

```
%Problem 23:
%Part a:
syms y;
solve(y-y^3==0,y)
% y=0 is the only real rool when alpha is less than or equal to 1 since the
% other two become imaginary due to taking the sqaure root of a
% negative number. When alpha is greater than 1 there is three disticnt
% real roots since only square roots of posive numbers are taken.
```

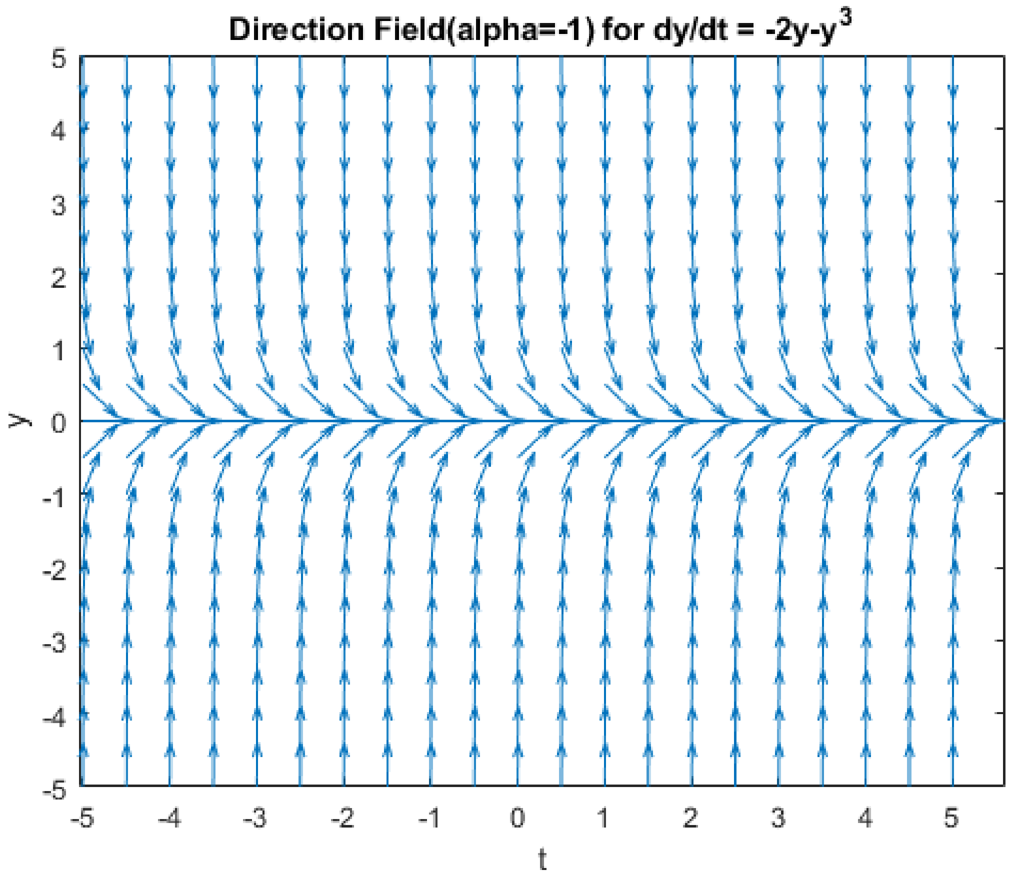
```
ans =

 -1
  0
  1
```

```
%Part b:
[t,y] = meshgrid(-5:0.5:5,-5:0.5:5);
dy = -y-y.^3;
dt = ones(size(dy));
dyu = dy./sqrt(dt.^2+dy.^2);
dtu = dt./sqrt(dt.^2+dy.^2);
quiver(t,y,dtu,dyu);
axis tight;
xlabel('t');
ylabel('y');
title('Direction Field(alpha=0) for dy/dt = -y-y^3');
hold off
%The only equilibrium solution exists for y=0, alpha=0. It is stable.
```

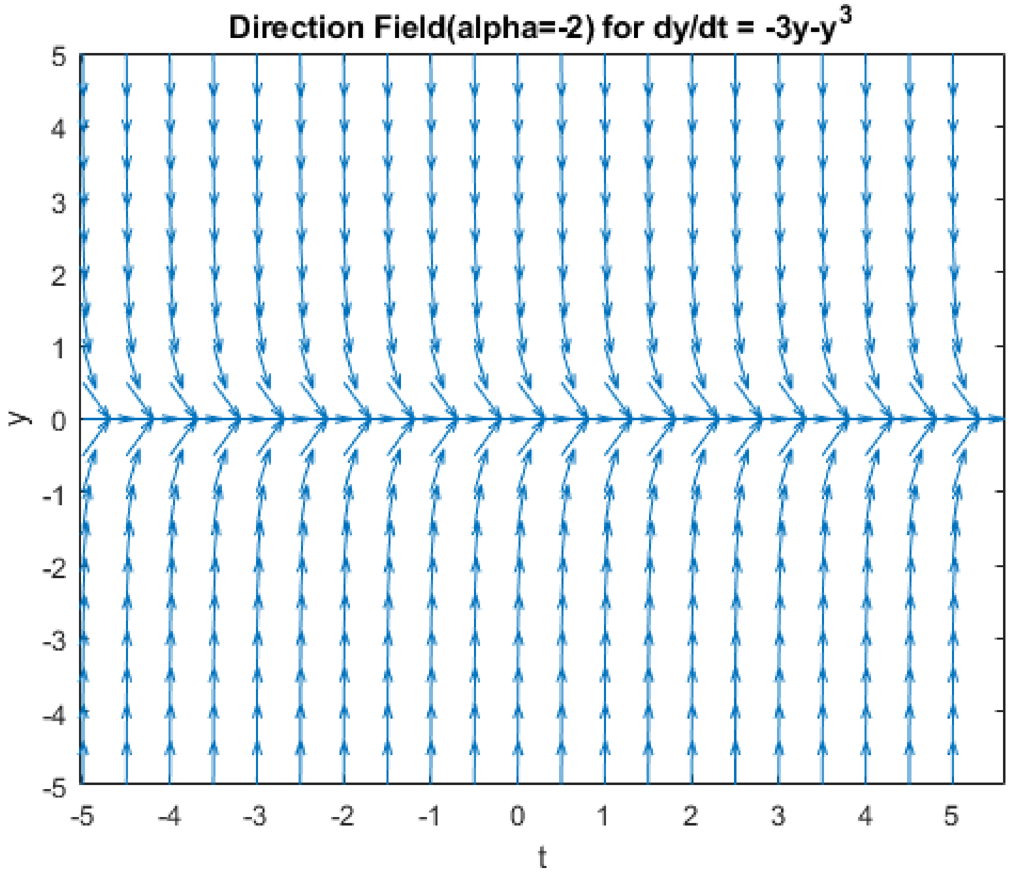



```
[t,y] = meshgrid(-5:0.5:5,-5:0.5:5);
dy = -2*y-y.^3;
dt = ones(size(dy));
dyu = dy./sqrt(dt.^2+dy.^2);
dtu = dt./sqrt(dt.^2+dy.^2);
```

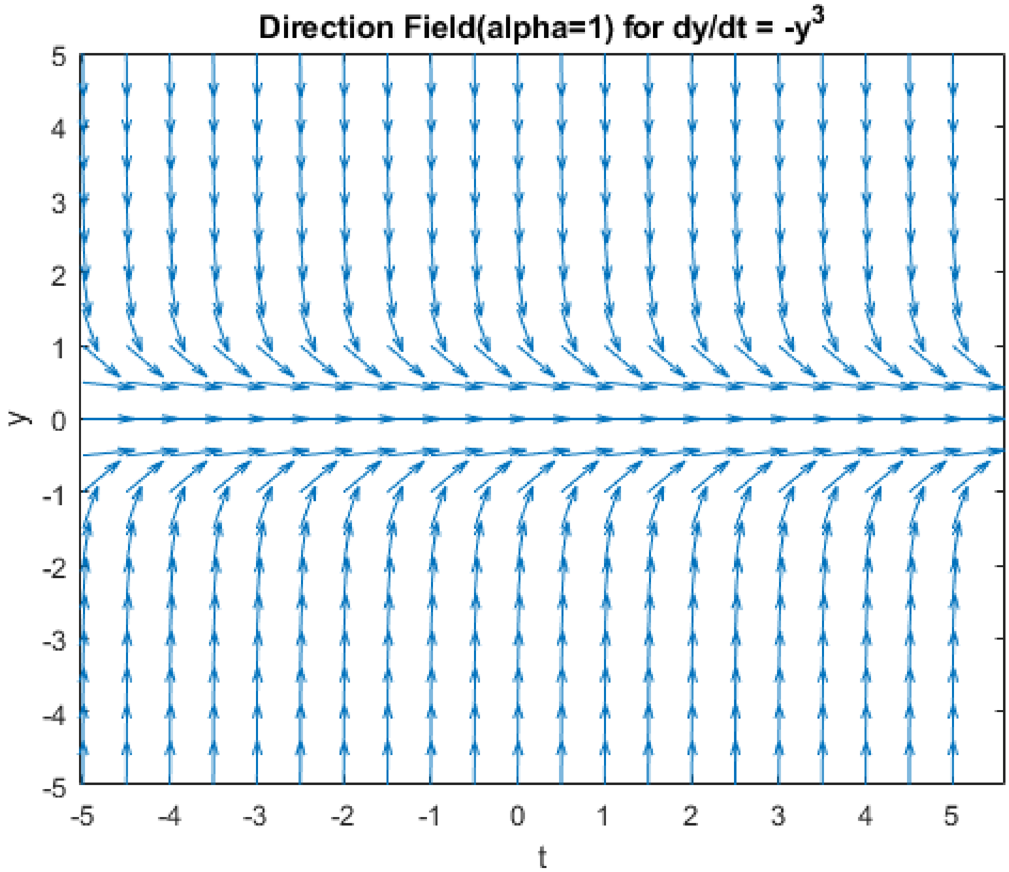
```
quiver(t,y,dtu,dyu);
axis tight;
xlabel('t');
ylabel('y');
title('Direction Field(alpha=-1) for dy/dt = -2y-y^3');
hold off
%The only equilibrium solution exists for y=0, alpha=0. It is stable.
```



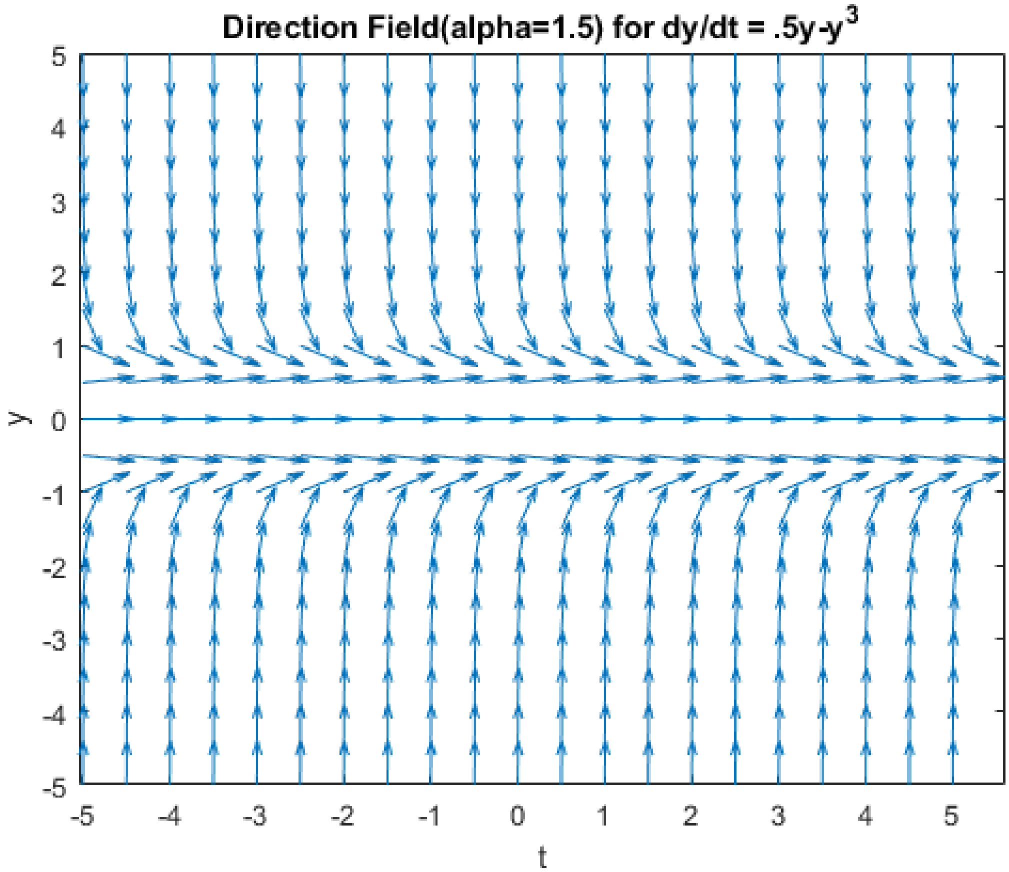

```
[t,y] = meshgrid(-5:0.5:5,-5:0.5:5);
dy = -3*y-y.^3;
dt = ones(size(dy));
dyu = dy./sqrt(dt.^2+dy.^2);
dtu = dt./sqrt(dt.^2+dy.^2);
quiver(t,y,dtu,dyu);
axis tight;
xlabel('t');
ylabel('y');
title('Direction Field(alpha=-2) for dy/dt = -3y-y^3');
hold off
%The only equilibrium solution exists for y=0, alpha=0. It is stable.
```



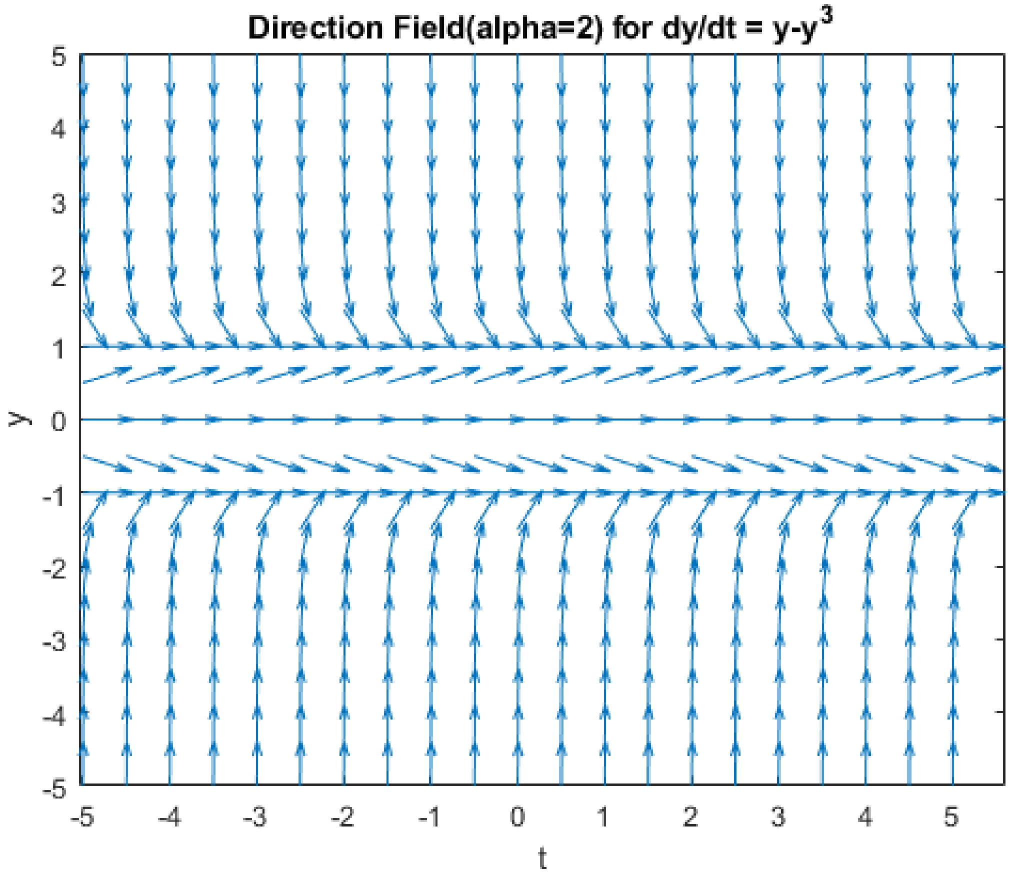

```
%Part c:
[t,y] = meshgrid(-5:0.5:5,-5:0.5:5);
dy = -y.^3;
dt = ones(size(dy));
dyu = dy./sqrt(dt.^2+dy.^2);
dtu = dt./sqrt(dt.^2+dy.^2);
quiver(t,y,dtu,dyu);
axis tight;
xlabel('t');
ylabel('y');
title('Direction Field(alpha=1) for dy/dt = -y^3');
hold off
%The equilibrium solution exists for y=0,alpha=1. It is unstable.
```




```
%Part d
[t,y] = meshgrid(-5:0.5:5,-5:0.5:5);
dy = .5*y-y.^3;
dt = ones(size(dy));
dyu = dy./sqrt(dt.^2+dy.^2);
dtu = dt./sqrt(dt.^2+dy.^2);
quiver(t,y,dtu,dyu);
axis tight;
xlabel('t');
ylabel('y');
title('Direction Field(alpha=1.5) for dy/dt = .5y-y^3');
hold off
%The equilibrium solution exists for y=-.5,0,.5.They are stable, unstable,
%stable.
```




```
[t,y] = meshgrid(-5:0.5:5,-5:0.5:5);
dy = y-y.^3;
dt = ones(size(dy));
dyu = dy./sqrt(dt.^2+dy.^2);
dtu = dt./sqrt(dt.^2+dy.^2);
quiver(t,y,dtu,dyu);
axis tight;
xlabel('t');
ylabel('y');
title('Direction Field(alpha=2) for dy/dt = y-y^3');
hold off
%The equilibrium solution exists for y=-1,0,1.They are stable, unstable,
%stable.
```




```
%Part e:
%As alpha increases through 1,the number of equilibrium solutions
%changes from 1 to 2. Thus the stable solution y=0 bifurcates into
%2 stable solutions.
```

Published with MATLAB® R2017b