In [2]:
```
from sympy import *
```

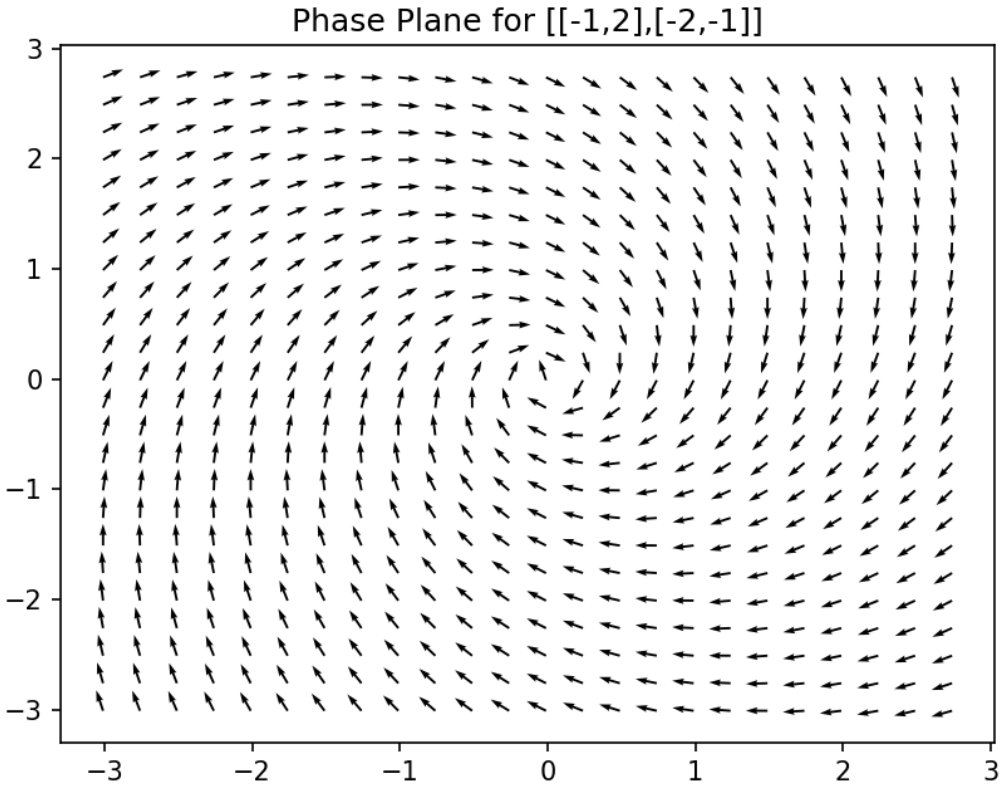
Example 2: x' = [[-1, 2], [-2, -1]] x

In [3]:
```
t=symbols('t')
x1=Function('x1')
x2=Function('x2')
deq1=diff(x1(t),t)+x1(t)-2*x2(t)
deq2=diff(x2(t),t)+2*x1(t)+x2(t)
dsolve([deq1,deq2])
```

Out[3]:
```
[Eq(x1(t), (2*C1*cos(2*t) + 2*C2*sin(2*t))*exp(-t)),
 Eq(x2(t), (-2*C1*sin(2*t) + 2*C2*cos(2*t))*exp(-t))]
```

In [4]:
```
matplotlib notebook
```

In [5]:
```
# phase plane
import numpy as np
import matplotlib.pyplot as plt
X1, X2 = np.meshgrid(np.arange(-3.01, 2.99, .25), np.arange(-3.01, 2.99, .2
5)) # adjust domain and range and spacing as needed
X1p = -X1+2*X2
X2p = -2*X1-X2
#Normalize the arrows by dividing by their magnitude (focus on direction)
U=1/(X1p**2+X2p**2)**(0.5)*X1p
V=1/(X1p**2+X2p**2)**(0.5)*X2p
plt.figure()
plt.title('Phase Plane for [[-1,2],[-2,-1]]')
Q = plt.quiver(X1, X2, U, V) # draws the arrows at (X,Y) with slope dYdX
print('Notice how the paths spiral into the origin (spiral sink)')
```




```
Notice how the paths spiral into the origin (spiral sink)
```

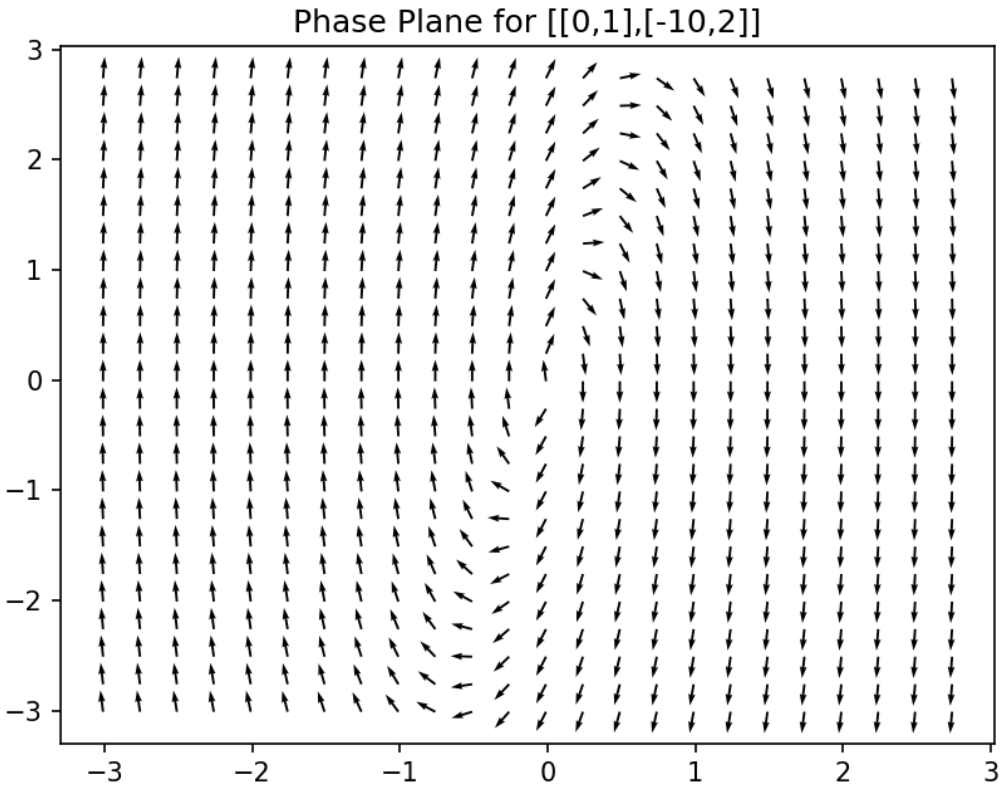
Example 2: x' = [[0, 1], [-10, 2]]x

In [6]:
```
t=symbols('t')
x1=Function('x1')
x2=Function('x2')
deq1=diff(x1(t),t)-x2(t)
deq2=diff(x2(t),t)+10*x1(t)-2*x2(t)
dsolve([deq1,deq2])
```

Out[6]:
```
[Eq(x1(t), (C1*cos(3*t) + C2*sin(3*t))*exp(t)),
 Eq(x2(t), (C1*(-3*sin(3*t) + cos(3*t)) + C2*(sin(3*t) + 3*cos(3*t)))*exp
(t))]
```

In [7]:
```
matplotlib notebook
```

In [8]:
```
X1, X2 = np.meshgrid(np.arange(-3.01, 2.99, .25), np.arange(-3.01, 2.99, .2
5)) # adjust domain and range and spacing as needed
X1p = X2
X2p = -10*X1+2*X2
#Normalize the arrows by dividing by their magnitude (focus on direction)
U=1/(X1p**2+X2p**2)**(0.5)*X1p
V=1/(X1p**2+X2p**2)**(0.5)*X2p
plt.figure()
plt.title('Phase Plane for [[0,1],[-10,2]]')
Q = plt.quiver(X1, X2, U, V) # draws the arrows at (X,Y) with slope dYdX
print('Paths spiral away from the origin (spiral source)')
```




```
Paths spiral away from the origin (spiral source)
```

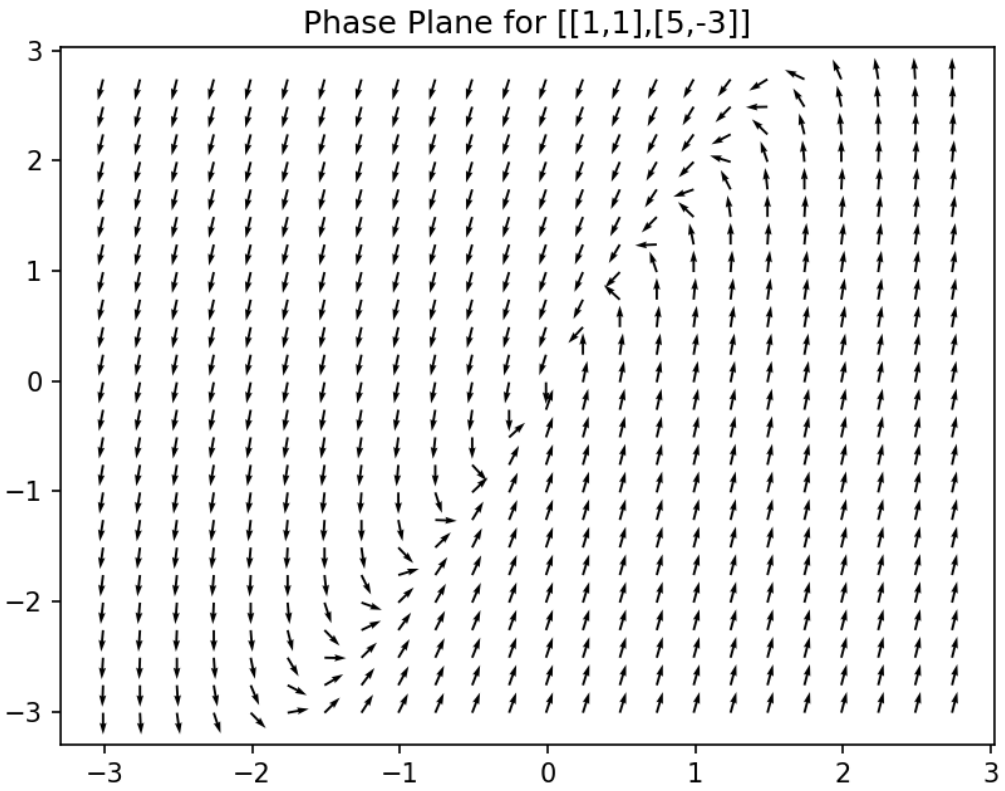
Example 3: x' = [[1 -1], [5 -3]]x

In [9]:
```
t=symbols('t')
x1=Function('x1')
x2=Function('x2')
deq1=diff(x1(t),t)-x1(t)+x2(t)
deq2=diff(x2(t),t)-5*x1(t)+3*x2(t)
dsolve([deq1,deq2])
```

Out[9]:
```
[Eq(x1(t), (-C1*cos(t) - C2*sin(t))*exp(-t)),
 Eq(x2(t), (C1*(-sin(t) - 2*cos(t)) + C2*(-2*sin(t) + cos(t)))*exp(-t))]
```

In [10]:
```
matplotlib notebook
```

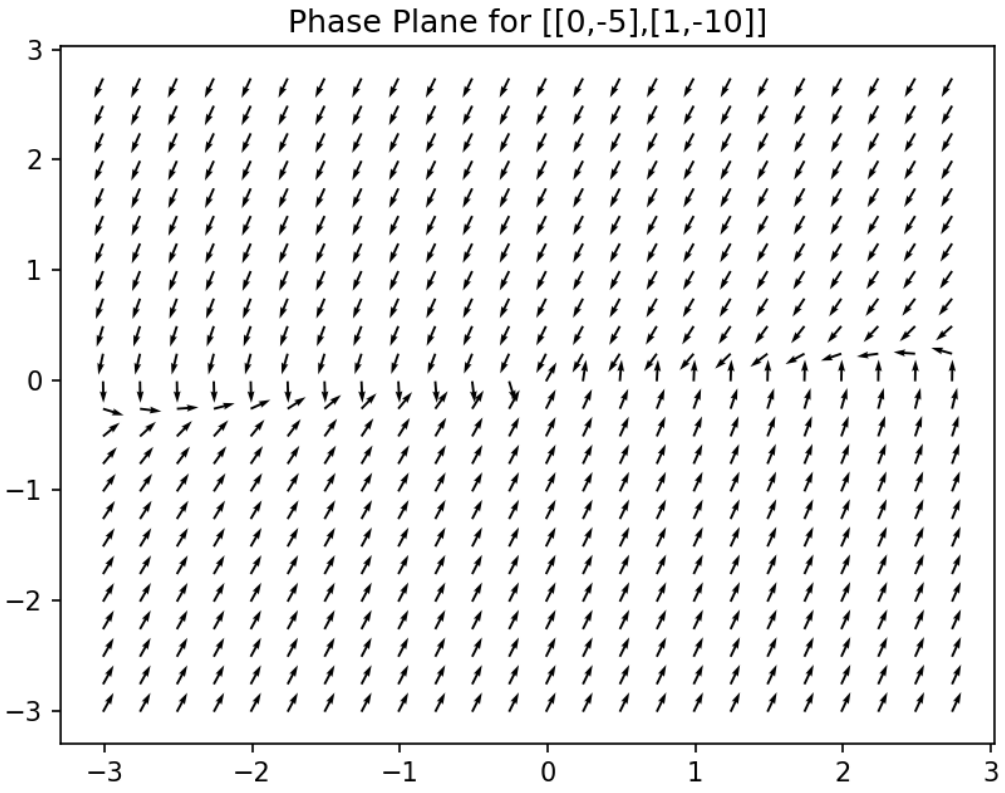
In [11]:
```
X1, X2 = np.meshgrid(np.arange(-3.01, 2.99, .25), np.arange(-3.01, 2.99, .2
5)) # adjust domain and range and spacing as needed
X1p = X1-X2
X2p = 5*X1-3*X2
#Normalize the arrows by dividing by their magnitude (focus on direction)
U=1/(X1p**2+X2p**2)**(0.5)*X1p
V=1/(X1p**2+X2p**2)**(0.5)*X2p
plt.figure()
plt.title('Phase Plane for [[1,1],[5,-3]]')
Q = plt.quiver(X1, X2, U, V) # draws the arrows at (X,Y) with slope dYdX
```



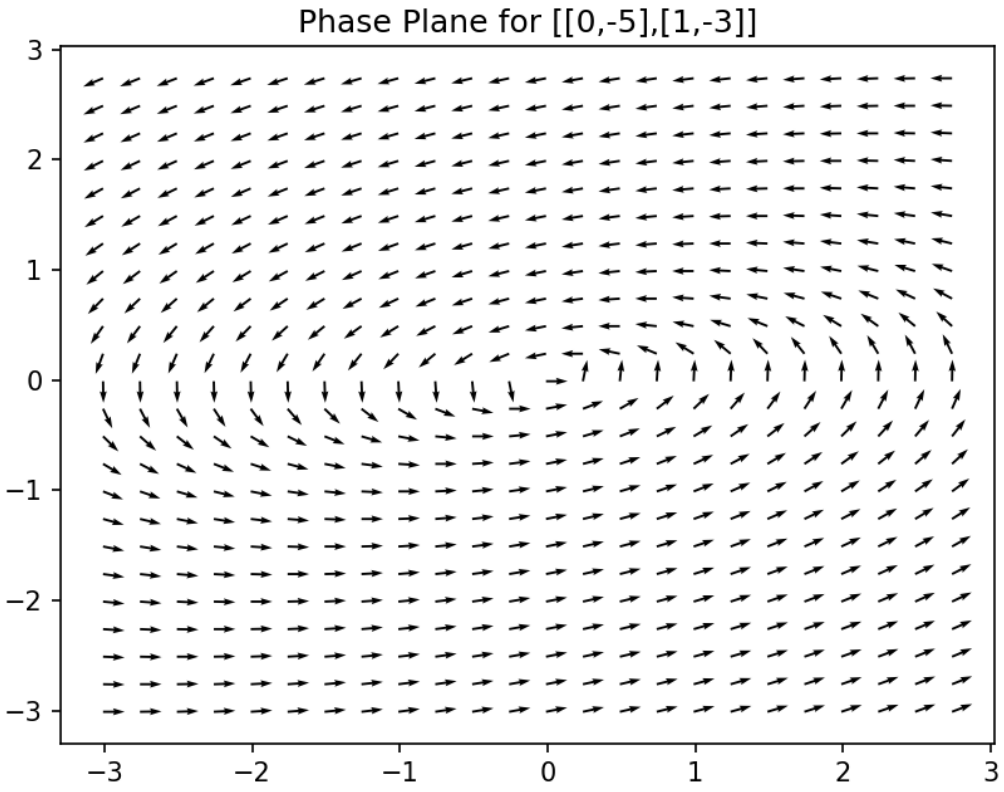
Example 4: x' = [[0, -5], [1 a]]x (Examples of phase plots in different intervals shown)

In [9]:
```
matplotlib notebook
```

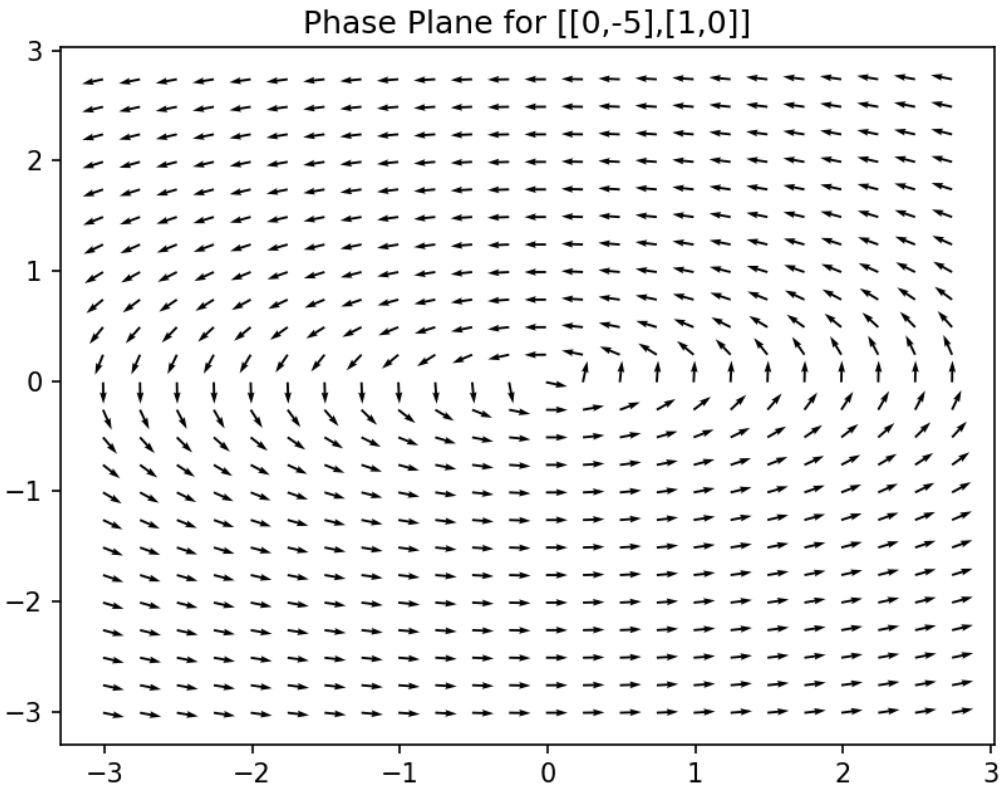
In [12]:
```
# Examples of phase planes
# a = -10
import numpy as np
import matplotlib.pyplot as plt
X1, X2 = np.meshgrid(np.arange(-3.01, 2.99, .25), np.arange(-3.01, 2.99, .2
5)) # adjust domain and range and spacing as needed
X1p = -5*X2
X2p = X1-10*X2
#Normalize the arrows by dividing by their magnitude (focus on direction)
U=1/(X1p**2+X2p**2)**(0.5)*X1p
V=1/(X1p**2+X2p**2)**(0.5)*X2p
plt.figure()
plt.title('Phase Plane for [[0,-5],[1,-10]]')
Q = plt.quiver(X1, X2, U, V) # draws the arrows at (X,Y) with slope dYdX
```



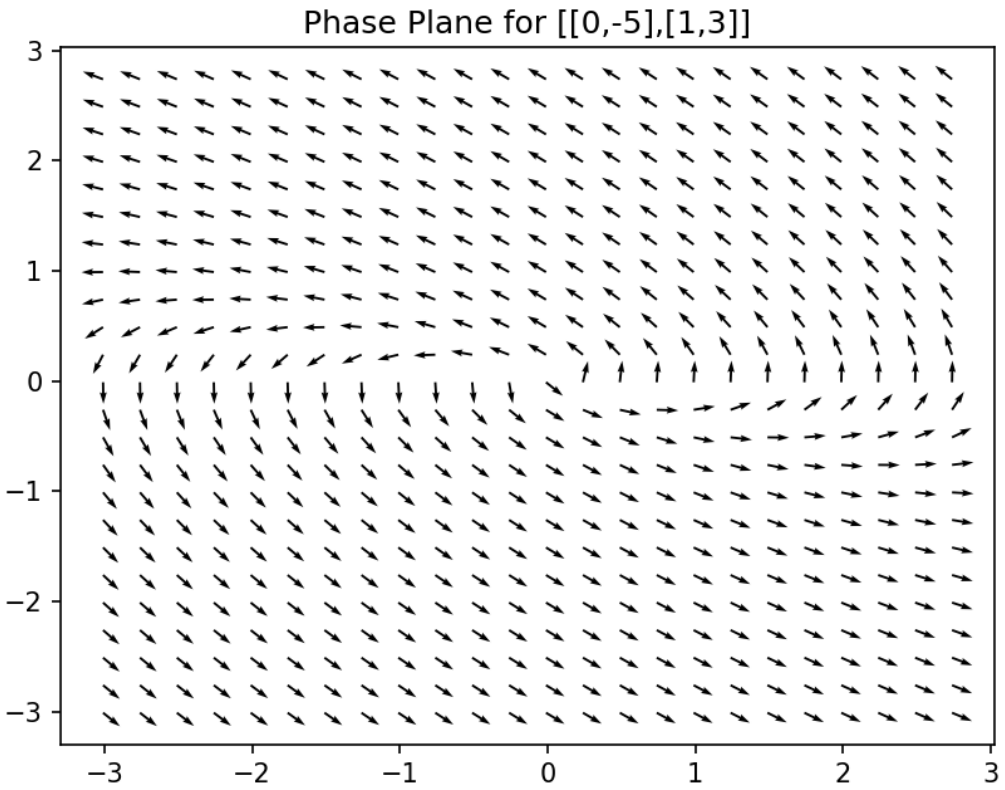
In [13]:
```
# a = -3
import numpy as np
import matplotlib.pyplot as plt
X1, X2 = np.meshgrid(np.arange(-3.01, 2.99, .25), np.arange(-3.01, 2.99, .2
5)) # adjust domain and range and spacing as needed
X1p = -5*X2
X2p = X1-X2
#Normalize the arrows by dividing by their magnitude (focus on direction)
U=1/(X1p**2+X2p**2)**(0.5)*X1p
V=1/(X1p**2+X2p**2)**(0.5)*X2p
plt.figure()
plt.title('Phase Plane for [[0,-5],[1,-3]]')
Q = plt.quiver(X1, X2, U, V) # draws the arrows at (X,Y) with slope dYdX
```



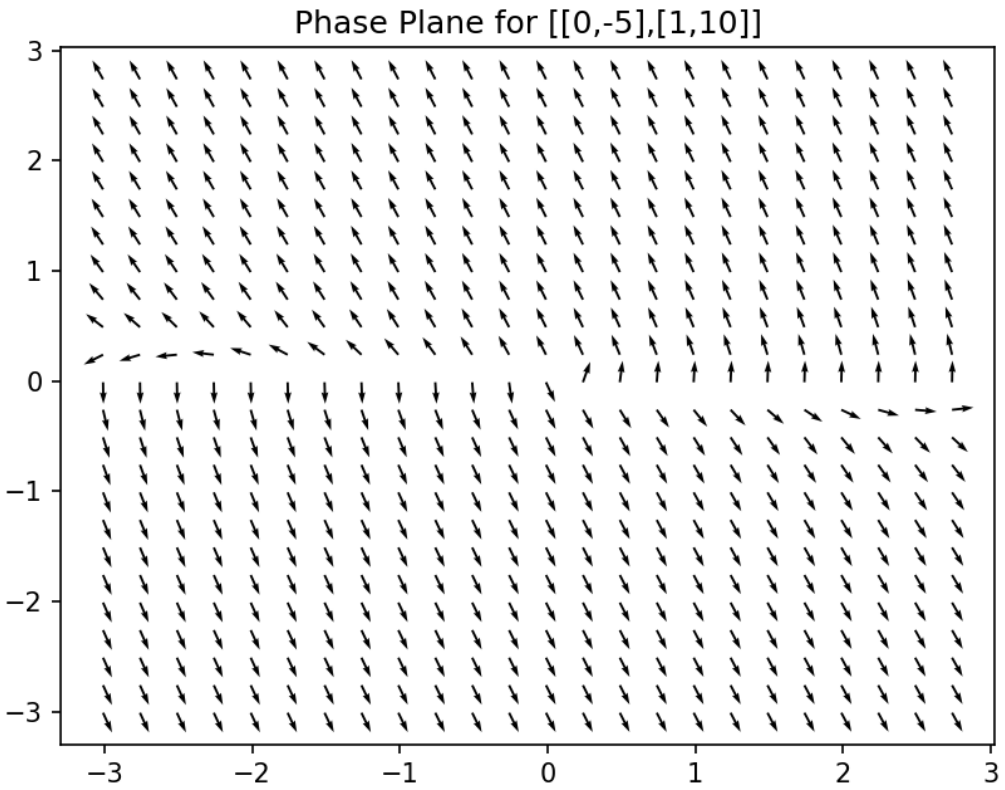
In [14]:
```
# a = 0
import numpy as np
import matplotlib.pyplot as plt
X1, X2 = np.meshgrid(np.arange(-3.01, 2.99, .25), np.arange(-3.01, 2.99, .2
5)) # adjust domain and range and spacing as needed
X1p = -5*X2
X2p = X1
#Normalize the arrows by dividing by their magnitude (focus on direction)
U=1/(X1p**2+X2p**2)**(0.5)*X1p
V=1/(X1p**2+X2p**2)**(0.5)*X2p
plt.figure()
plt.title('Phase Plane for [[0,-5],[1,0]]')
Q = plt.quiver(X1, X2, U, V) # draws the arrows at (X,Y) with slope dYdX
```



In [15]:
```
# a = 3
import numpy as np
import matplotlib.pyplot as plt
X1, X2 = np.meshgrid(np.arange(-3.01, 2.99, .25), np.arange(-3.01, 2.99, .2
5)) # adjust domain and range and spacing as needed
X1p = -5*X2
X2p = X1+3*X2
#Normalize the arrows by dividing by their magnitude (focus on direction)
U=1/(X1p**2+X2p**2)**(0.5)*X1p
V=1/(X1p**2+X2p**2)**(0.5)*X2p
plt.figure()
plt.title('Phase Plane for [[0,-5],[1,3]]')
Q = plt.quiver(X1, X2, U, V) # draws the arrows at (X,Y) with slope dYdX
```



In [16]:
```
# a = 10
import numpy as np
import matplotlib.pyplot as plt
X1, X2 = np.meshgrid(np.arange(-3.01, 2.99, .25), np.arange(-3.01, 2.99, .2
5)) # adjust domain and range and spacing as needed
X1p = -5*X2
X2p = X1+10*X2
#Normalize the arrows by dividing by their magnitude (focus on direction)
U=1/(X1p**2+X2p**2)**(0.5)*X1p
V=1/(X1p**2+X2p**2)**(0.5)*X2p
plt.figure()
plt.title('Phase Plane for [[0,-5],[1,10]]')
Q = plt.quiver(X1, X2, U, V) # draws the arrows at (X,Y) with slope dYdX
```



In [ ]: