In [1]:
```
from sympy import *
```

Example 1: x' = [[-3, 0], [0,-1]]x

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

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

Example 2: x' = [[3, -2], [2, -2]]x

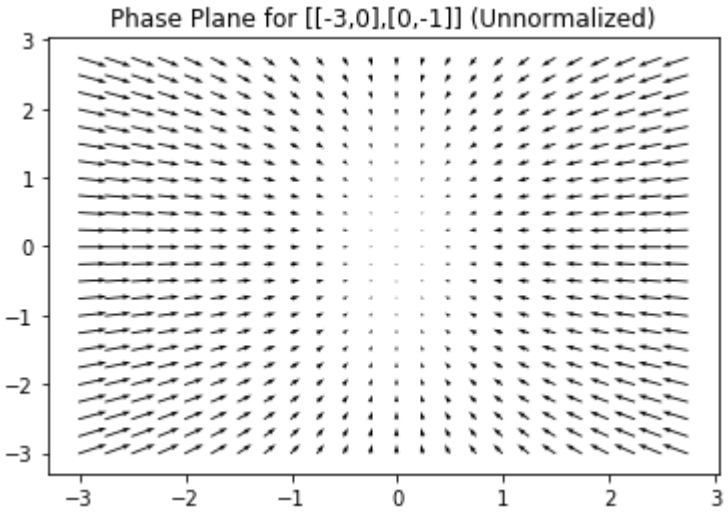
In [3]:
```
t=symbols('t')
x1=Function('x1')
x2=Function('x2')
deq1=diff(x1(t),t)-3*x1(t)+2*x2(t)
deq2=diff(x2(t),t)-2*x1(t)+2*x2(t)
dsolve([deq1,deq2])
```

Out[3]:
```
[Eq(x1(t), -2*C1*exp(2*t) - 2*C2*exp(-t)),
 Eq(x2(t), -C1*exp(2*t) - 4*C2*exp(-t))]
```

Graph of phase plane in x1-x2 plane

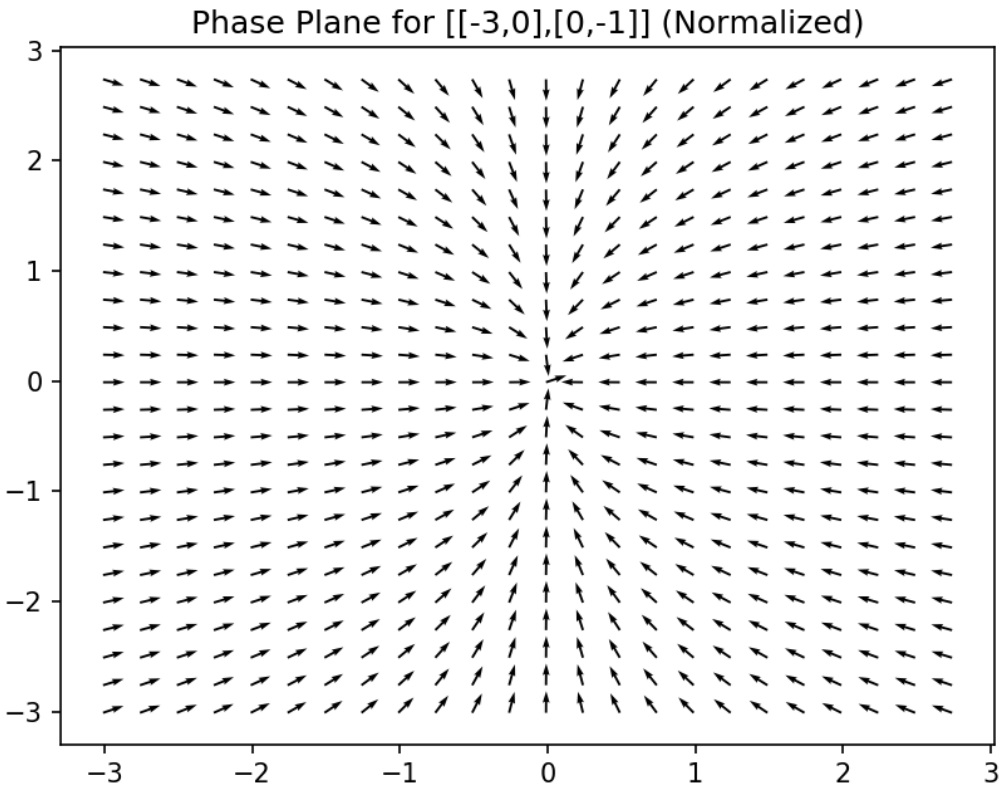
In [16]:
```
matplotlib notebook
```

In [4]:
```
import numpy as np  #NOTE:  Since we already imported from sympy, we will use a shortcut prefix for numpy
import matplotlib.pyplot as plt
X1, X2 = np.meshgrid(np.arange(-3.01, 2.99, .25), np.arange(-3.01, 2.99, .25)) # adjust domain and range and spacing as needed
X1p = -3*X1    # X' = [ [-3 0],[0 -1]]X
X2p = -1*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 [[-3,0],[0,-1]] (Unnormalized)')
Q = plt.quiver(X1, X2, X1p,X2p) # draws the unnormalized arrows at (X,Y) with slope dYdX
```



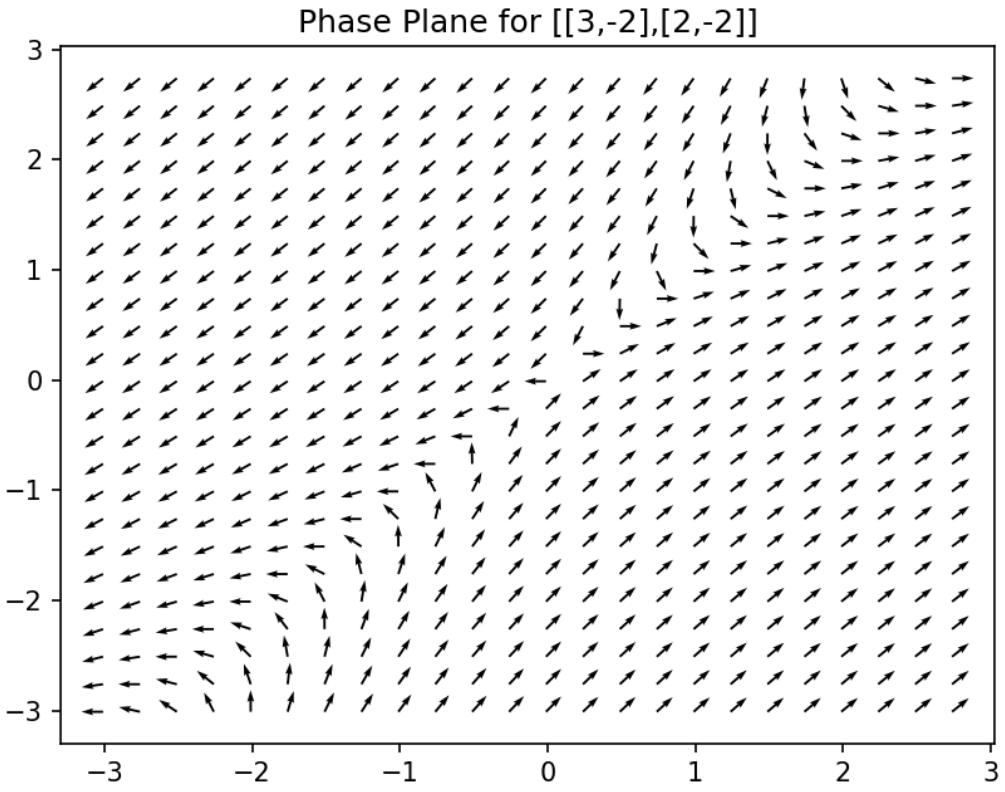
In [5]:
```
matplotlib notebook
```

In [6]:
```
Qn = plt.quiver(X1, X2, U, V) # draws the normalized arrows at (X,Y) with slope dYdX
plt.title('Phase Plane for [[-3,0],[0,-1]] (Normalized)')
```



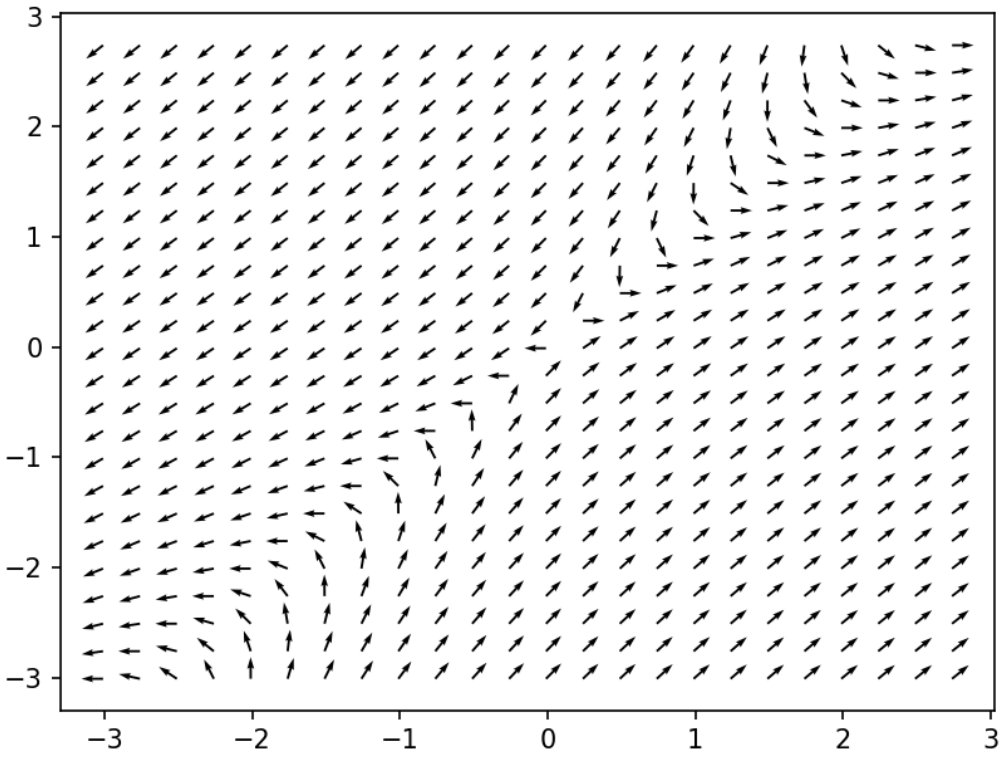
Out[6]: Text(0.5, 1.0, 'Phase Plane for [[-3,0],[0,-1]] (Normalized)')

In [7]:
```
X1, X2 = np.meshgrid(np.arange(-3.01, 2.99, .25), np.arange(-3.01, 2.99, .25)) # adjust domain and range and spacing as needed
X1p = 3*X1-2*X2
X2p = 2*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 [[3,-2],[2,-2]]')
Q = plt.quiver(X1, X2, U, V) # draws the unnormalized arrows at (X,Y) with slope dYdX
```



In [22]:
```
matplotlib notebook
```

In [23]:
```
Qn = plt.quiver(X1, X2, nU, nV) # draws the normalized arrows at (X,Y) with slope dYdX
```



In [ ]:
```

```