```python
from astropy import units as u
from astropy import constants as const
from matplotlib import pyplot as plt
import numpy as np
import matplotlib as mpl

# Planet properties
Mp = 0.6 * const.M_jup
Rp = 1.3 * const.R_jup
a = (0.05 * u.au).to(u.m)
e = 0.17
# Star properties
Ms = 1.1 * const.M_sun
Rs = 1.1 * const.R_sun
# System Orientation
i = 88 * np.pi / 180
f = 30 * np.pi / 180
# constants
G = const.G

p = a*(1-e*e)
h = np.sqrt(G*(Ms+Mp)*p)
r = p/(1+e*np.cos(f))
b = r*np.cos(i)/Rs
Vsky = (h/p)*(1+e*np.cos(f))
d = 2*np.sqrt((Rs+Rp)**2-(b*Rs)**2)
t = d/Vsky
T = np.sqrt(4*(np.pi)**2*a**3/(G*(Ms+Mp)))

tlist = np.linspace(-t,t,10000)
dsplist = np.array([(Rs*np.sqrt(b**2+(Vsky*tp/Rs)**2)).value for tp in tlist])*u.m
def A(d):
    if d >= Rs+Rp:
        return 0 * u.m * u.m
```

```
35      elif d <= Rs-Rp:
36          return np.pi*Rp**2
37      else:
38          theta_p = np.arccos((Rs**2+d**2-Rp**2
    )/(2*Rs*d)).value
39          theta_s = np.arccos((Rp**2+d**2-Rs**2
    )/(2*Rp*d)).value
40          a = theta_s*Rp**2+theta_p*Rs**2-Rs*d*
    np.sin(theta_p)
41          return a
42  L = np.array([(1-A(dsp)/(np.pi*Rs**2)).value
    for dsp in dsplist])
43
44  fig, ax = plt.subplots(figsize=(8,4), layout=
    'constrained')
45  ax.plot(tlist.value/3600, L, color='blue', lw
    =2)
46  ax.axhline(y=1-Rp**2/Rs**2, color='black', lw
    =1, ls='--')
47  ax.text(0.53*t.value/3600, np.min(L)+0.0005,
    'eclipse depth: {:.4f}'.format(1-np.min(L)),
    color='black', size=12)
48  ax.axvline(x=-0.5*t.value/3600, ymax=(1-np.
    min(L)+0.001)/(1-np.min(L)+0.003), color='
    black', lw=1, ls='--')
49  ax.axvline(x=0.5*t.value/3600, ymax=(1-np.min
    (L)+0.001)/(1-np.min(L)+0.003), color='black'
    , lw=1, ls='--')
50  ax.text(0, 1.0005, 'eclipse takes {:.2f}
    hours of {:.2f} days orbit'.format(t.to(u.
    hour).value, T.to(u.day).value), color='black
    ', size=13, ha='center', va='bottom')
51  ax.set_ylim(np.min(L)-0.001, 1.002)
52  ax.set_xlabel('time [hour]',fontsize='xx-
    large')
```

```
53 ax.set_ylabel('Normalized Flux',fontsize='xx-
   large')
54 ax.xaxis.set_minor_locator(mpl.ticker.
   MultipleLocator(0.2))
55 ax.yaxis.set_minor_locator(mpl.ticker.
   MultipleLocator(0.001))
56 fig.savefig('RV.pdf', format='pdf', dpi=600,
   bbox_inches='tight', pad_inches=0.2)
57
58
59
```