Using frame transformations to get to AltAz

To actually do anything with observability we need to convert to a frame local to an on-earth observer. By far the most common choice is horizontal coordinates, or "AltAz" coordinates. We first need to specify both where and when we want to try to observe.

In [32]:
```
from astropy.coordinates import EarthLocation
from astropy.time import Time

observing_location = EarthLocation(lat='31d57.5m', lon='-111d35.8m', height=2096*u.m)  # Kitt Peak, Arizona
# If you're using astropy v1.1 or later, you can replace the above with this:
#observing_location = EarthLocation.of_site('Kitt Peak')

observing_time = Time('2010-12-21 1:00')  # 1am UTC=6pm AZ mountain time
```

Now we use these to create an `AltAz` frame object. Note that this frame has some other information about the atmosphere, which can be used to correct for atmospheric refraction. Here we leave that alone, because the default is to ignore this effect (by setting the pressure to 0).

In [33]:
```
from astropy.coordinates import AltAz

aa = AltAz(location=observing_location, obstime=observing_time)
aa
```

Out[33]:
```
<AltAz Frame (obstime=2010-12-21 01:00:00.000, location=(-1994310.09211632, -5037908.6
06337594, 3357621.752122168) m, pressure=0.0 hPa, temperature=0.0 deg_C, relative_humi
dity=0, obswl=1.0 micron)>
```

Now we can just transform our ICRS `SkyCoord` to `AltAz` to get the location in the sky over Kitt Peak at the requested time.

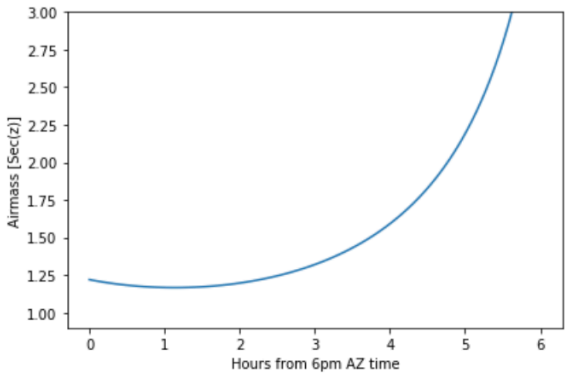
In [34]:
```
hcg7_center.transform_to(aa)
```

Out[34]:
```
<SkyCoord (AltAz: obstime=2010-12-21 01:00:00.000, location=(-1994310.09211632, -50379
08.606337594, 3357621.752122168) m, pressure=0.0 hPa, temperature=0.0 deg_C, relative_
humidity=0, obswl=1.0 micron): (az, alt) in deg
    ( 149.19392032,  55.06247359)>
```

Alright, it's up at 6pm, but that's pretty early to be observing. We could just try various times one at a time to see if the airmass is at a darker time, but we can do better: lets try to create an airmass plot.

In [35]:
```
# this gives a Time object with an *array* of times
delta_hours = np.linspace(0, 6, 100)*u.hour
full_night_times = observing_time + delta_hours
full_night_aa_frames = AltAz(location=observing_location, obstime=full_night_times)
full_night_aa_coos = hcg7_center.transform_to(full_night_aa_frames)

plt.plot(delta_hours, full_night_aa_coos.secz)
plt.xlabel('Hours from 6pm AZ time')
plt.ylabel('Airmass [Sec(z)]')
plt.ylim(0.9,3)
plt.tight_layout()
```



Great! Looks like it's at the lowest airmass in another hour or so (7pm). But might that might still be twilight... When should we start observing for proper dark skies? Fortunately, astropy provides a `get_sun` function that can be used to check this. Lets use it to check if we're in 18-degree twilight or not.