```
r""" This module is designed to determine a multitude of data points
for a given orbit

Inputs Vary

Functions:
    Level 0:
        Unit Vector
    Level 1:
        Angular Momentum
        Eccentricity
        Specific Mechanical Energy
    Level 2:
        Line of Nodes
        Semi Latus Rectum
        True Anomaly
        Inclination
    Level 3:
        Semi Major Axis
        Right Ascension of the Ascending Node
        Argument of Periapsis
        Radius of Periapsis
        Radius of Apoapsis
        Flight Path Angle
    Level 4:
        Period

Author: Thomas J Susi       GWU         tsusi13@gwu.edu
"""

import numpy as np
import matplotlib.pyplot as plt
from astro import constants

r"""Level 0 Functions for Unit Vectors, etc.
"""
def unit_vector(vector):
    r"""Computes the Unit Vector of the given vector

    Inputs:
    vector :(vector) Given in any components with any units

    Outputs:
    vector_hat :(vector) Given in any components but is unitless

    Author: Thomas J Susi       GWU         tsusi13@gwu.edu
    """

    vector_hat = vector / (np.linalg.norm(vector))
```

```
    return vector_hat


r"""Level 1 Functions for Angular Momentum, Eccentricity, and Specific
Energy
"""
def ang_momentum(r_vector, v_vector):
    r"""Computes the Angular Momentum of the system

    Inputs:
    r_vector :(vector) Given in components of ijk with units in km
    v_vector :(vector) Given in components of ijk with units in km/s

    Outputs:
    h_vector :(vector) Given in components of ijk with units in kmkm/s

    Author: Thomas J Susi       GWU         tsusi13@gwu.edu
    """

    h_vector = np.cross(r_vector, v_vector)

    return h_vector

def eccentricity(mu, r_vector, v_vector, h_vector):
    r"""Computes the Eccentricity of the system

    Inputs:
    mu :(scalar) Given in units kmkmkm/ss
    r_vector :(vector) Given in components of ijk with units in km
    v_vector :(vector) Given in components of ijk with units in km/s
    h_vector :(vector) Given in components of ijk with units in kmkm/s


    Outputs:
    e_vector :(vector) Given in components ijk but is unitless

    Author: Thomas J Susi       GWU         tsusi13@gwu.edu
    """

    e_vector = np.divide(np.cross(v_vector, h_vector), mu) -
np.divide(r_vector, np.linalg.norm(r_vector))

    return e_vector

def spec_mech_energy(mu, r_vector, v_vector):
    r"""Computes the Specific Mechanical Energy of the system

    Inputs:
    mu :(scalar) Given in units kmkmkm/ss
    r_vector :(vector) Given in components of ijk with units in km
```

```
    v_vector :(vector) Given in components of ijk with units in km/s

    Outputs:
    sme :(scalar) Given in units kmkm/ss

    Author: Thomas J Susi       GWU         tsusi13@gwu.edu
    """

    sme = ((np.linalg.norm(v_vector) * np.linalg.norm(v_vector)) / 2)
- (mu / np.linalg.norm(r_vector))

    return sme


r"""Level 2 Functions for Line of Nodes, True Anomaly, Inclination,
and Semi_Latus_Rectum
"""
def line_of_nodes(h_vector):
    r"""Computes the Line of Nodes of the system

    Inputs:
    h_hat :(unit vector) Given in components of ijk but is unitless

    Outputs:
    n_vector :(vector) Given in components of ijk in km

    Author: Thomas J Susi       GWU         tsusi13@gwu.edu
    """

    k_hat = [0,0,1]

    n_vector = np.cross(k_hat, h_vector)

    return n_vector

def semi_latus_rectum(mu, h_vector):
    r"""Computes the Line of Nodes of the system

    Inputs:
    mu :(scalar) Given in units kmkmkm/ss
    h_vector :(vector) Given in components of ijk with units in kmkm/s

    Outputs:
    p :(scalar) Given in units km

    Author: Thomas J Susi       GWU         tsusi13@gwu.edu
    """

    p = ((np.linalg.norm(h_vector))**2) / mu
```

```
    return p

def true_anom(r_vector, h_vector, e_vector):
    r"""Computes the Line of Nodes of the system

    Inputs:
    r_vector :(vector) Given in components of ijk with units in km
    h_vector :(vector) Given in components of ijk with units in kmkm/s
r_hat :(unit vector) Given in components of ijk but is unitless
    e_vector :(vector) Given in components of ijk but is unitless

    Outputs:
    theta :(angle) Given in units of radians

    Author: Thomas J Susi       GWU         tsusi13@gwu.edu
    """

    theta = np.arctan2(np.dot(h_vector, np.cross(e_vector,
r_vector)),np.linalg.norm(h_vector) * np.dot(e_vector, r_vector))

    return theta

def inclination(h_hat):
    r"""Computes the Line of Nodes of the system

    Inputs:
    h_hat :(unit vector) Given in components of ijk but is unitless

    Outputs:
    inc :(angle) Given in units of radians

    Author: Thomas J Susi       GWU         tsusi13@gwu.edu
    """

    k_hat = [0,0,1]

    inc = np.arccos(np.dot(k_hat, h_hat))

    return inc


r"""Level 3 Functions for Semi-Major Axis, RAAN, Argument of
Periapsis, Radius of Periapsis, Radius of Apoapsis, and Flight Path
Angle
"""
def semi_major_axis(p, e):
    r"""Computes the Semi-Major Axis of the system

    Inputs:
    p :(scalar) Given in units of km
```

```
    e :(scalar) Given but is unitless

    Outputs:
    a :(scalar) Given in units of km

    Author: Thomas J Susi       GWU         tsusi13@gwu.edu
    """

    a = p / (1 - e**2)

    return a

def R_A_A_N(n_vector):
    r"""Computes the RAAN of the system

    Inputs:
    n_vector :(vector) Given in components of ijk in km

    Outputs:
    raan :(angle) Given in units of radians

    Author: Thomas J Susi       GWU         tsusi13@gwu.edu
    """

    i_hat = [1,0,0]
    j_hat = [0,1,0]

    raan = np.arctan2(np.dot(j_hat, n_vector),np.dot(i_hat, n_vector))

    return raan

def arg_of_periapsis(n_vector, e_vector, h_vector):
    r"""Computes the Argument of Periapsis of the system

    Inputs:
    n_vector :(vector) Given in components of ijk in km
    e_vector :(vector) Given in components of ijk but is unitless
    h_vector :(vector) Given in components of ijk with units in kmkm/s

    Outputs:
    arg_peri :(angle) Given in units of radians

    Author: Thomas J Susi       GWU         tsusi13@gwu.edu
    """

    arg_peri = np.arctan2(np.dot(h_vector, np.cross(n_vector,
e_vector)),  np.linalg.norm(h_vector)*np.dot(n_vector, e_vector))

    return arg_peri
```

```
def rad_peri(p, e):
    r"""Computes the Radius of Periapsis of the system

    Inputs:
    p :(scalar) Given in units of km
    e :(scalar) Given but is unitless

    Outputs:
    r_p :(angle) Given in units of radians

    Author: Thomas J Susi       GWU         tsusi13@gwu.edu
    """
    r_p = p / (1 + e)

    return r_p

def rad_apo(p, e):
    r"""Computes the Radius of Apoapsis of the system

    Inputs:
    p :(scalar) Given in units of km
    e :(scalar) Given but is unitless

    Outputs:
    r_a :(angle) Given in units of radians

    Author: Thomas J Susi       GWU         tsusi13@gwu.edu
    """
    r_a = p / (1 - e)

    return r_a

def flight_ang(e, theta):
    r"""Computes the Flight Path Angle of the system

    Inputs:
    e :(scalar) Given but is unitless
    theta :(angle) Given in units of radians

    Outputs:
    fpa :(angle) Given in units of radians

    Author: Thomas J Susi       GWU         tsusi13@gwu.edu
    """

    fpa = np.arctan((e * np.sin(theta))/(1 + (e * np.cos(theta))))

    return fpa
```

```
r"""Level 4 Functions for Period
"""
def Orbit_Period(mu, a):
    r"""Computes the Period of the system

    Inputs:
    mu :(scalar) Given in units kmkmkm/ss
    a :(scalar) Given in units of km

    Outputs:
    period :(scalar) Given in units s

    Author: Thomas J Susi       GWU         tsusi13@gwu.edu
    """

    period = 2 * np.pi * np.sqrt((a * a * a) / mu)

    return period
```