At certain time intervals determined by inputs that the "user" gives. We have to recalculate its position, velocity, and acceleration.

Set up initial conditions

The initial values that we have include PosX and PosY of the spacecraft, VelX and VelY of the spacecraft, and the radius and mass of Mercury. We also have the gravity constant and the time and its interval we want to measure.

initialize any arrays / do any initial calculations

There are four arrays we would like

==========

time: an array of times with shape (nt,).

acc: (acc_x, acc_y), an array of spacecraft accelerations with shape (2, nt).

vel: (vel_x, vel_y), an array of spacecraft velocities with shape (2, nt) .

pos: (pos_x, pos_y), an array of spacecraft positions with shape (2, nt).

==========

This can be done with this piece of code:

```
timeArray = np.arange(0, totalTime + 1, interval)

posArray = np.reshape(np.ones(len(timeArray) * 2) * np.nan,
(2, -1))

velArray = np.reshape(np.ones(len(timeArray) * 2) * np.nan,
(2, -1))

accArray = np.reshape(np.ones(len(timeArray) * 2) * np.nan,
(2, -1))
```

This sets up 4 arrays all of the length we want with the shape we want. The inital values for these arrays (pos, vel, acc) can be set up in the for loop

for each time step:

calculate acceleration

```
accArray[0, i], accArray[1, i] = grav_acc(posArray[0, i],
posArray[0, i], mercM)
```

calculate change in vel, position during the time step

```
def changeInPosition(vel, accel, dtime):

    return (vel * dtime) + (0.5 * accel * (dtime**2))

def changeInVelocity(accel, dtime):

    return accel/dtime
```

update vel and position

```
posArray[0, i] = posArray[0, b] +
changeInPosition(velArray[0, b], accArray[0, b], interval)

posArray[1, i] = posArray[1, b] +
changeInPosition(velArray[1, b], accArray[1, b], interval)

velArray[0, i] = velArray[0, b] +
changeInVelocity(accArray[0, b], interval)

velArray[1, i] = velArray[1, b] +
changeInVelocity(accArray[1, b], interval)
```

In [ ]: