**Simulating a simple re-entry trajectory (simple\_re-entry\_simulation):**

* Only considers the gravitational forces
* Dynamics are in the x-y plane and since there’s no initial velocity in the y-direction, it’s just a straight line

**Simulating a re-entry where the orbit is designed so that re-entry begins at apogee (re-entry\_at\_apogee):**

* Model a highly elliptical orbit
* Create an initial orbit where the satellite’s perigee is above Earth’s atmosphere

Rough plan:

1. Orbital parameters:

* Perigee: 300km (so r initial is 6771km)
* Apogee: 7000km

2. Initial orbit:

* Semi- major axis a=(r\_perigee + r\_apogee)/2
* to describe the size of the elliptical orbit
* longest distance from the centre of the ellipse to its edge
* larger a = longer orbit
* Eccentricity i.e. the elongation of the ellipse
* e = (r\_perigee - r\_apogee)/(r\_perigee + r\_apogee)

3. Initial conditions for re-entry

* initialise velocity and position at this point
* orbital mechanics equations
* initial velocity is 7.8km/s (typical speed for an object in LEO)

4. re-entry dynamics

* atmospheric drag
* gravitational force
* add drag into the dynamics once re-entry begins

**To include velocity plots (re-entry\_at\_apogee\_with\_velocity\_plot):**

1. define parameters

* Radius of Earth
* Gravitational parameter mu
* Initial altitude at apogee (400km)
* Eccentricity = 0.1 (slightly elliptical orbit)
* Eccentricity = 0 at a circular orbit and 1 at a parabolic orbit
* We need eccentricity 0<e<1 (typical for satellites)

2. compute the semi major axis

3. calculate initial distance and velocity at apogee

4. re-entry dynamics function

* Gravitational acceleration
* Atmospheric drag which is only applicable below 100km altitude
* Use exponential atmosphere model- suitable for low accuracy and quick simulations
* Approximate drag coefficient = 2.2
* Calculate total accelerations in x and y (gravity and drag)

5. time span for simulation

6. solve differential equation

* solve\_ivp to solve initial value problems of differential equation
* solution = solve\_ivp(reentry\_dynamics, t\_span, initial\_state, t\_eval=t\_eval)
* re-entry dynamics defines the equations of motion
* t\_span is the time range for the simulation
* initial\_state specifies the start position and velocity
* results extracted

7. find re-entry point

8. plot the trajectory

Explaining the graphs:

* Once the satellite gets below 100km, atmospheric drag acts on it so the drag force increases
* Drag force is proportional to velocity squared so the force strongly opposes the satellite’s motion and rapidly slows it down

**Improving the model above (re-entry\_at\_apogee\_with\_velocity\_plot2):**

* In the first model, drag only activated at a specific altitude
* Whereas the new model updates dynamically as the speed and altitude transitions within the atmosphere
* In the new model gravitational and drag forces are calculated at each time step based on the position and velocity so the simulation of how the velocity changes is more realistic
* The previous model didn’t simulate how the satellite gradually slows down by atmospheric drag
* However the problems with this model is that is models the velocity to increase and decrease over time, which isn’t accurate to what would actually happen