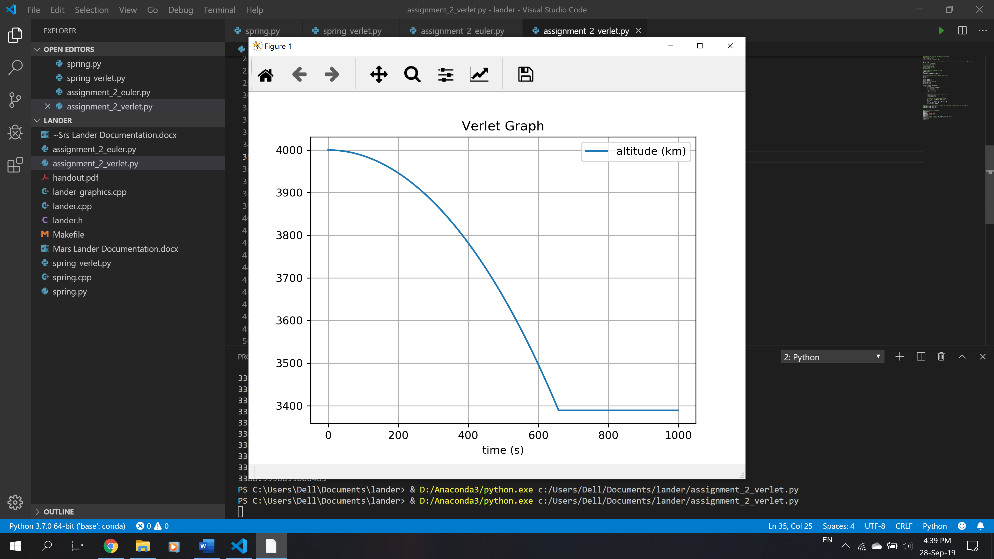
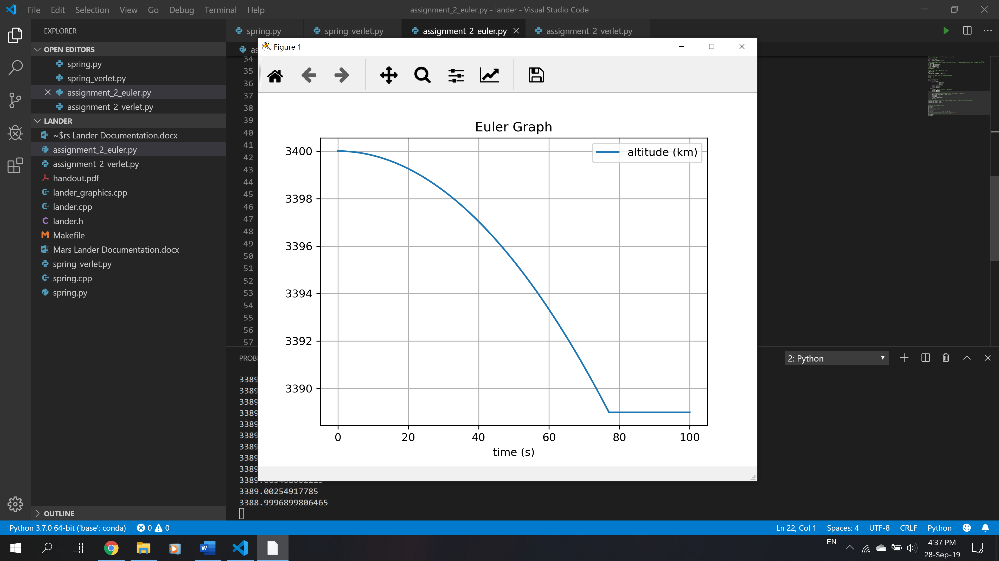
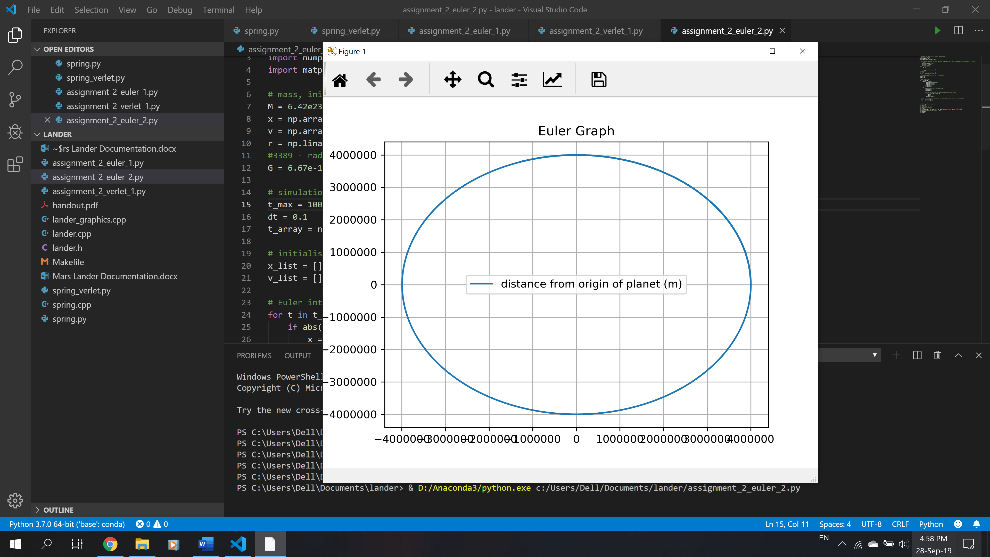
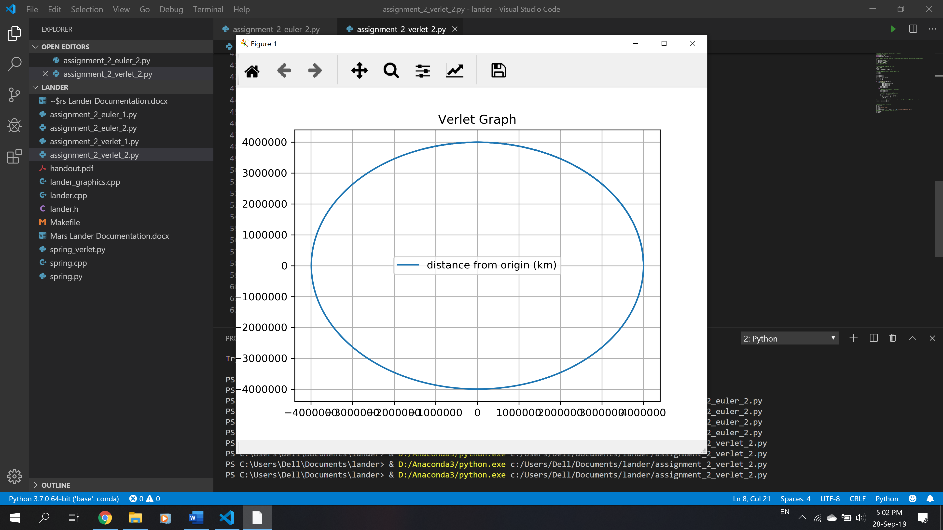
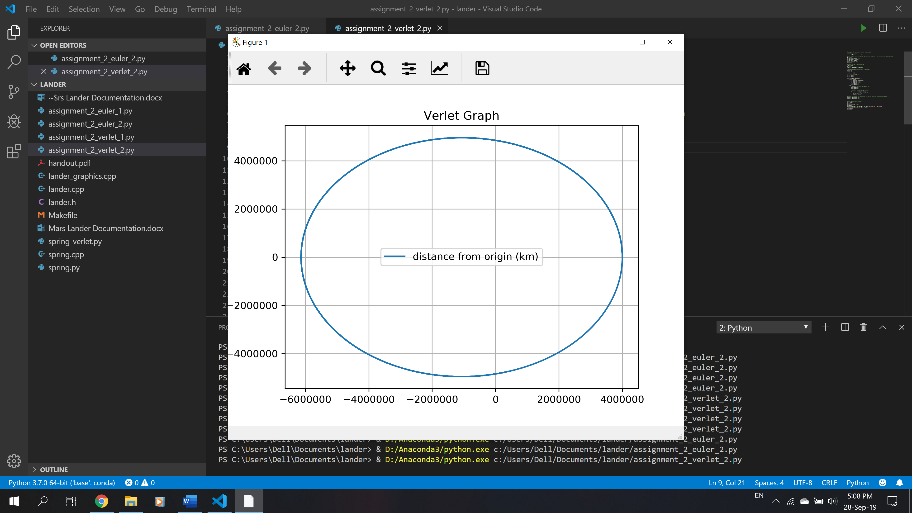
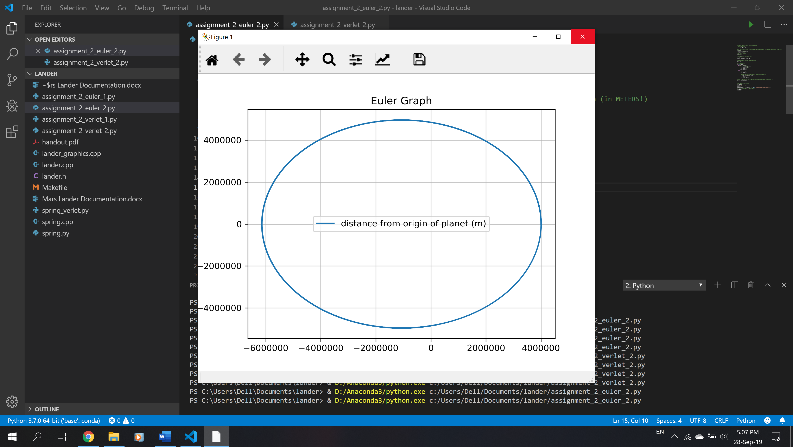
Assignment 1

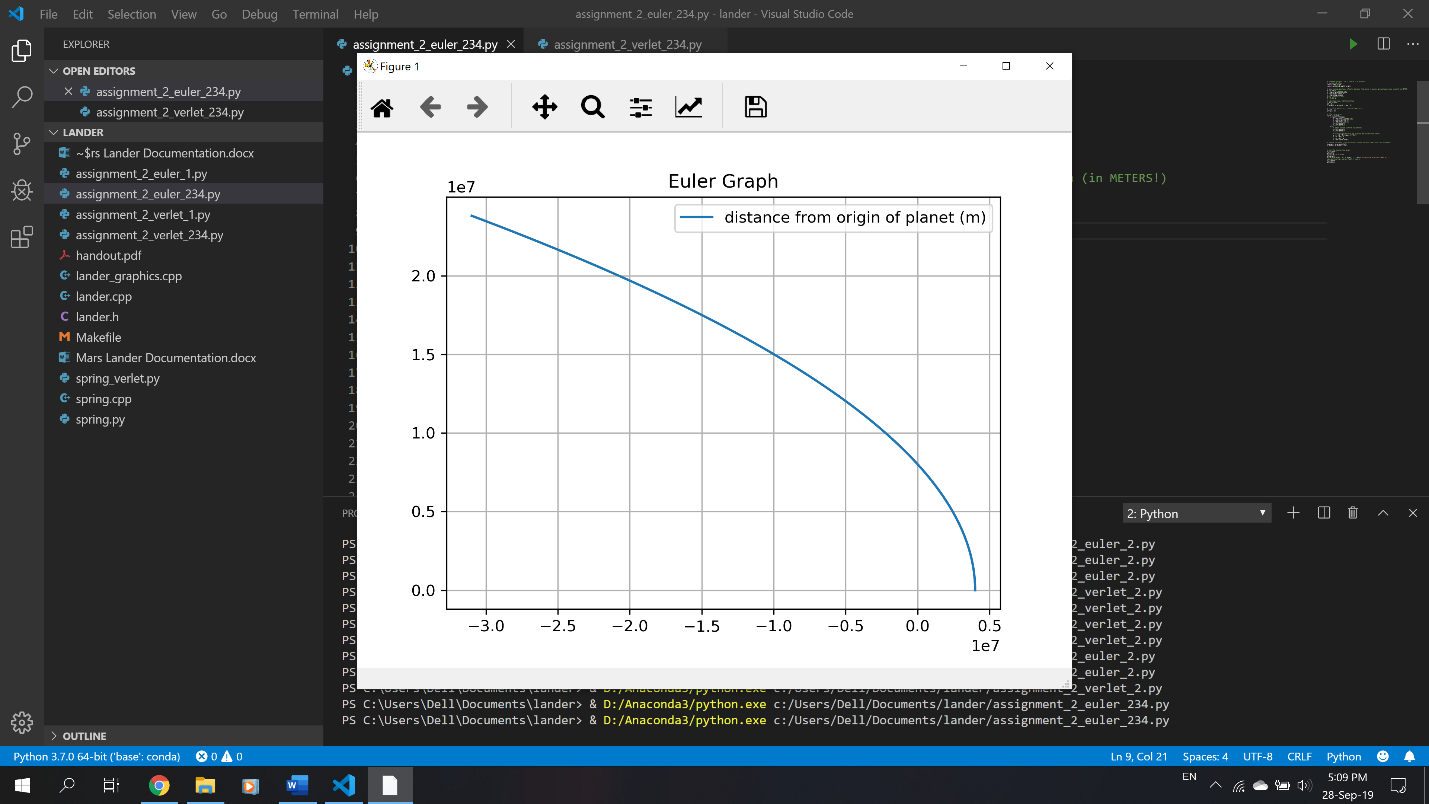
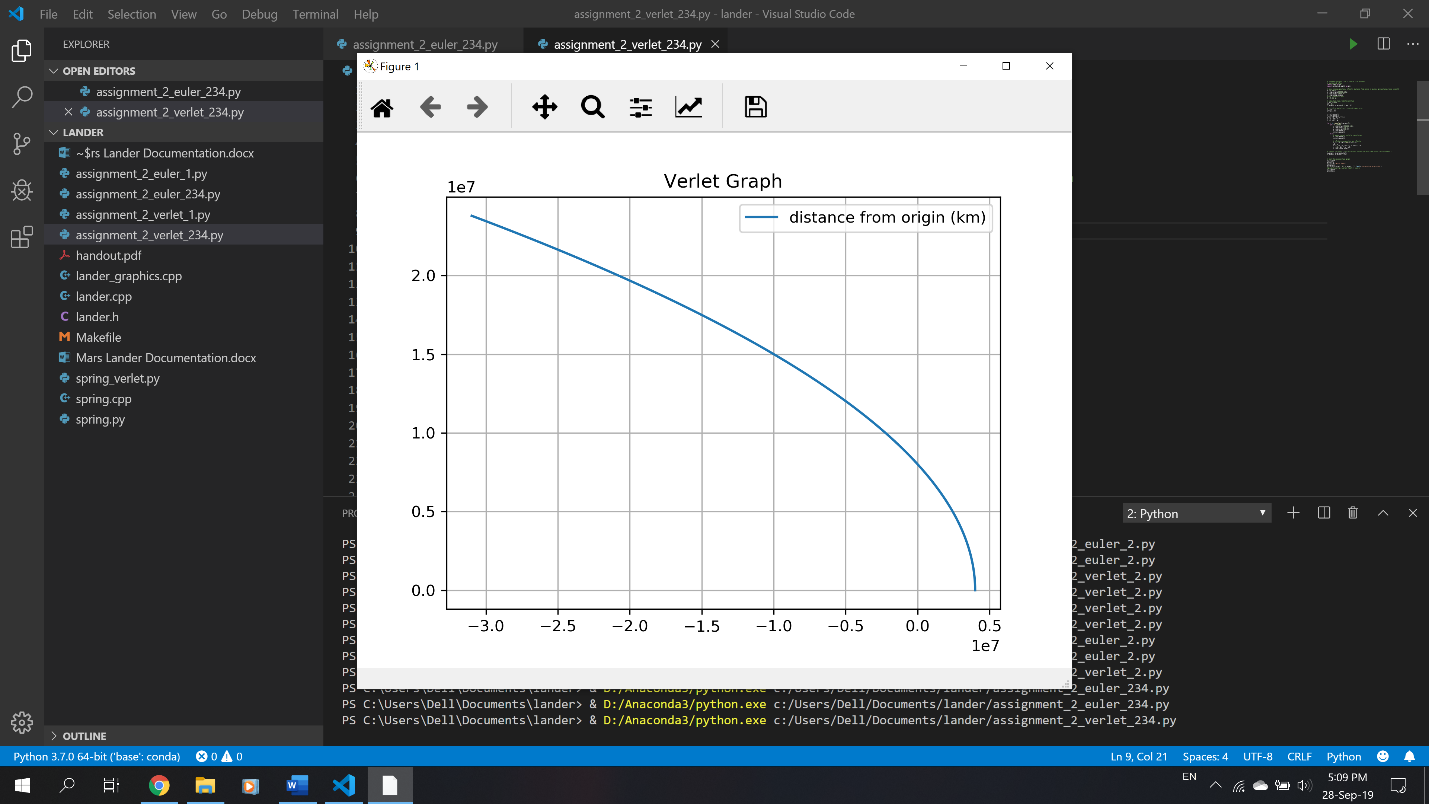
1. –
2. Verlet seems to be much more accurate compared to Euler (as Euler seems to always be diverging). As usual, if delta t decreases, the graph gets more accurate, but slower to compute.
3. 1.01 is the critical value for dt above which it becomes unstable.

Assignment 2

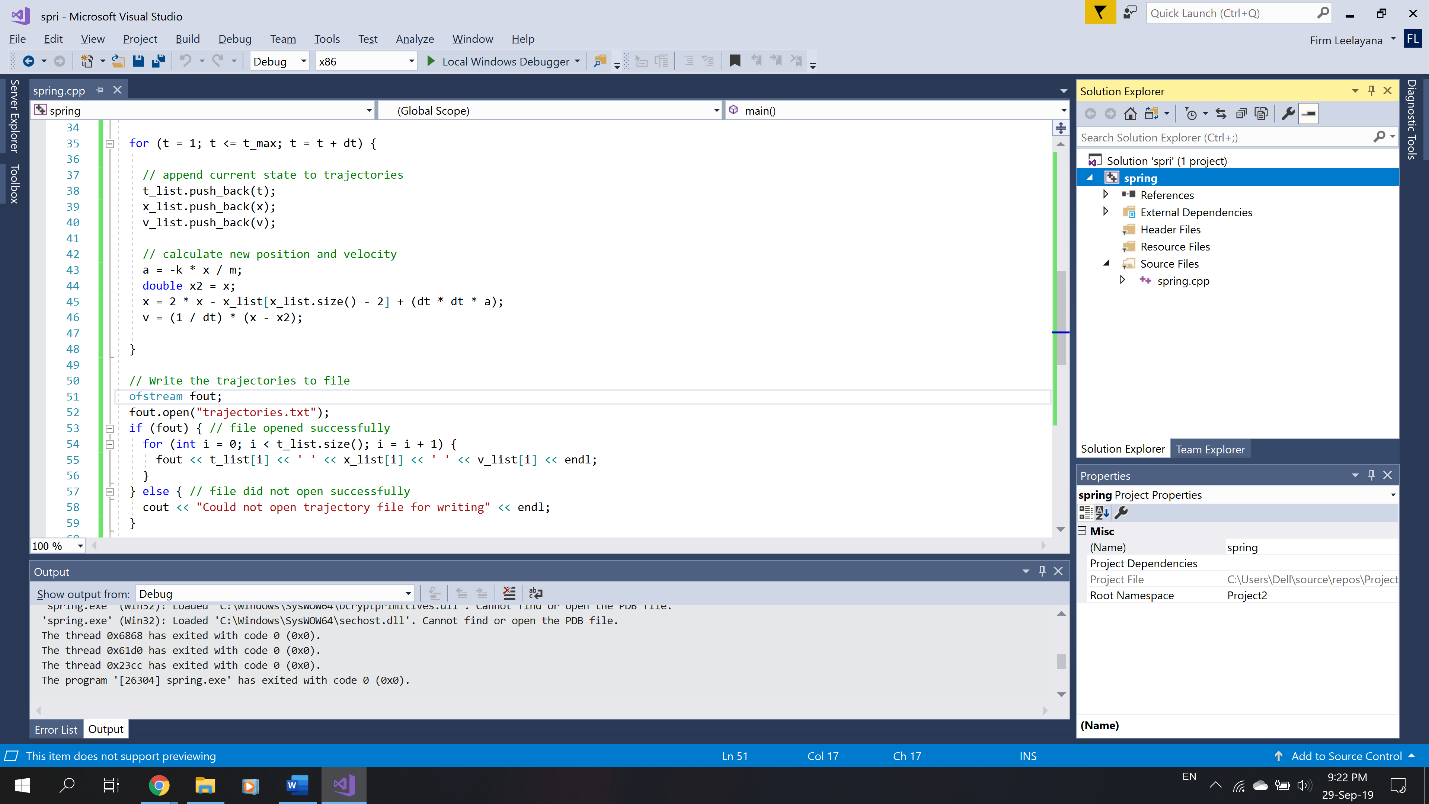
*For 4000km altitude, 3270 ms-1 is the speed for circular motion and escape velocity is 4627ms-1.*

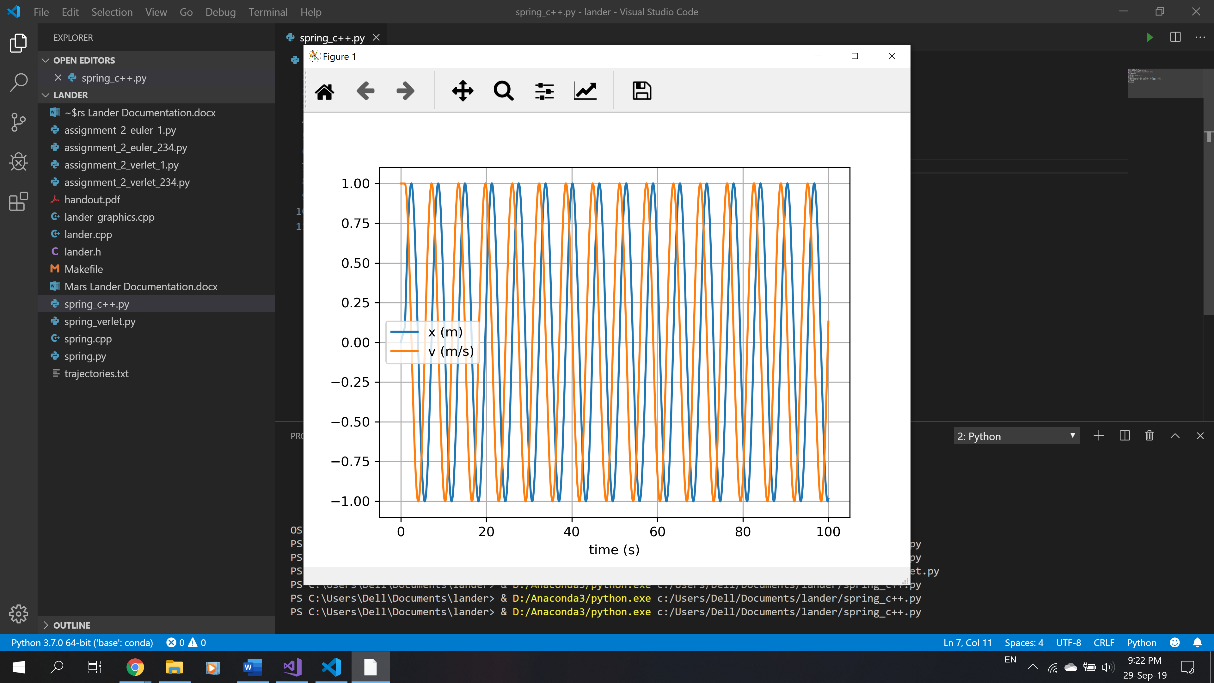
1. Straight down descent (from 3400 and 4000 km):
2. Circular Orbit
3. Elliptical Orbit



1. Hyperbolic escape

Assignment 3

* Python runs much slower compared to C++
* C++ compiler optimization runs faster than C++ unoptimized

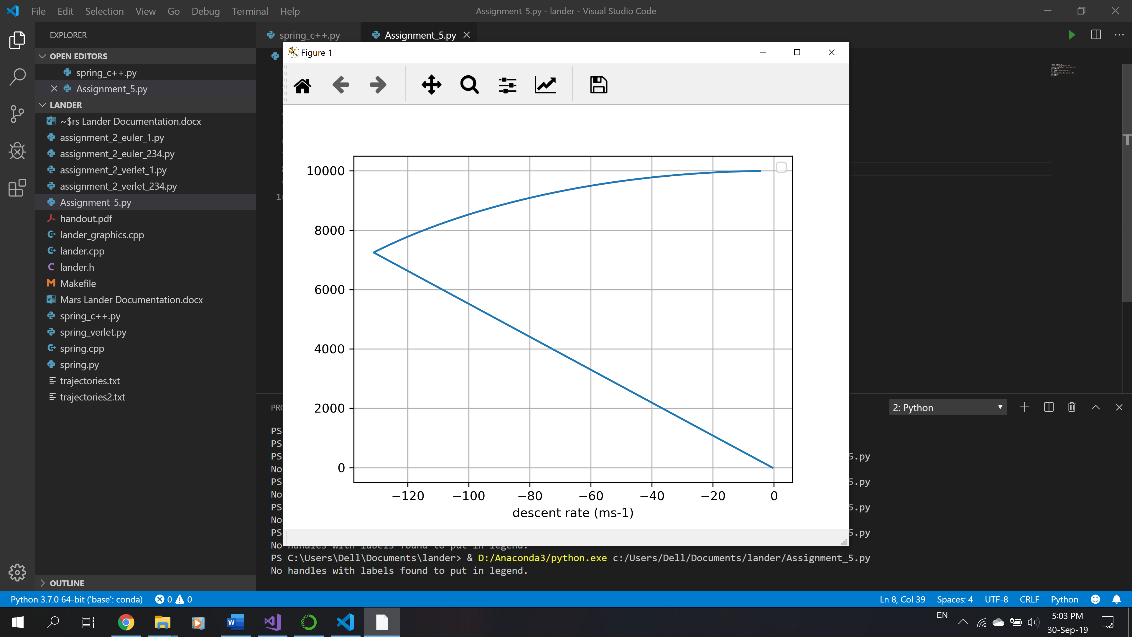


Assignment 4

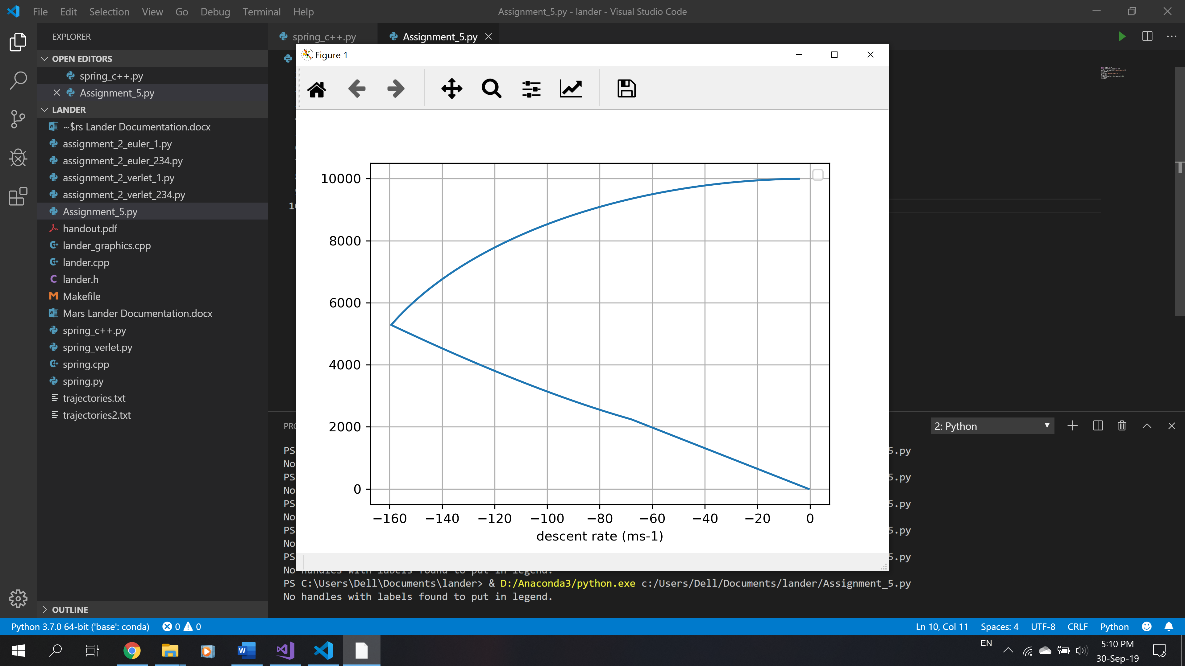
* Small discrepancies with the given Scenario cases, but overall correct.
* Both integrators perform very well, giving similar scenario solutions. However, with Scenario 4, the Euler integrator continues indefinitely whilst the Verlet integrator causes the lander to crash.

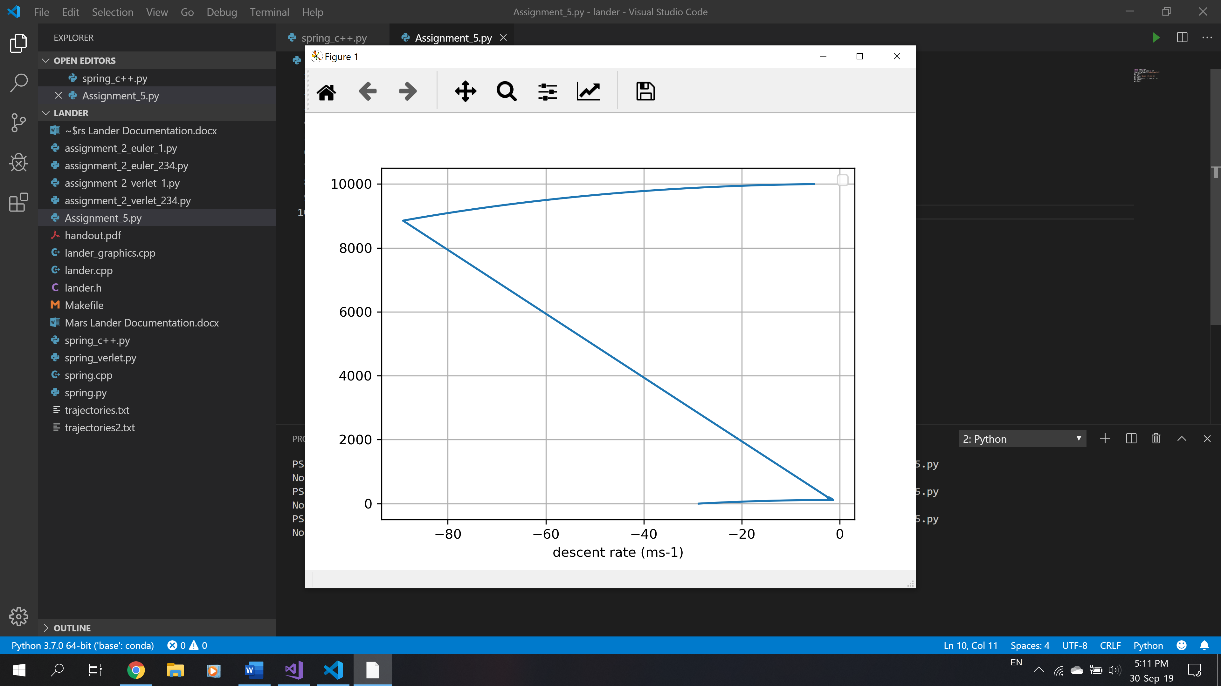
Assignment 5

* Kh too high = kh \* h term will be too large, and overshadow other terms easily until the lander nearly reaches the ground, by which there won’t be enough time for the lander thrust to decelerate the lander. (e will always be negative, so throttle won’t be on until the last segment)
* Kh too low = kh \* h term negligible compared to other terms, so throttle will always be on, therefore it will run out of fuel. (e will always be positive, so throttle will always be set on)

kh = 0.018

kh = 0.03



kh = 0.01