In the first task I created a function, *moon\_position*, which takes time, *t*, as a parameter. The distance between the Moon and the Earth, the time period, *T*, and the initial condition were given and were used to compute the Moon’s position relative to the Earth. The initial angle was given in degrees, which was converted into radians, then the change in the angle was calculated and incorporated into the equation to find the equation of the position of the Moon. I then tested to see if my function works with the, *test\_moon\_position.m*, and I get a graph which is an ellipse, that looks like a circle, which is 10\*(10^8) meters in width and 8\*(10^8) meters in height.

For the second task I created a function, *spaceship\_acceleration*, that took the position of the Moon and the position of the spacecraft as parameters. The universal gravitational constant, the mass of the Earth and the mass of the Moon were given. I used these values along with the equation provided in the powerpoint to calculate the acceleration of the spaceship. I then used the file *test\_spaceship\_acceleration.m* to test if my function works. The output on the console is *passed* which means that I have got the correct acceleration for the spaceship.

For the third task I wrote a function *spaceship\_motion* which takes time, *t*, and the position matrix, *[x1,x2,v1,v2]*, as parameters. This function returns a matrix that has velocity and acceleration as its elements, *[v1,v2,a1,a2]*. The function  *spaceship\_acceleration* was used to calculate the acceleration of the spaceship by taking the *moon\_position* function evaluated at time *t* and the spaceship position as parameters. I used the *test\_spaceship\_motion.m* to see if my function works and it does. When this script is executed I get a graph that has an ellipse around a red circle, which shows the motion of the spaceship relative to the Earth.

For the fourth task I used the initial conditions that were given in the powerpoint and modified the *X0* variable in the *test\_spaceship\_motion.m* file to go along with these conditions. Using the condition *v0 = [2.0 \* 10^3 ; 4.0 \* 10^3]* I get an ellipse which has Earth as one of its foci. Earth is at the left foci. Using the condition *v0 = [3.9 \* 10^3 ; 6.0 \* 10^3]* I receive a graph that has Earth at the origin and an curve that starts at *1.5 \* 10^7m* on the x-axis and has an increasing gradient. This curve stops at around *6 \* 10^^7m* on the y-axis. It does not make a full ellipse.

For the fifth task I created a function *euler* which took a function, *f(t)*, a time period, *t=[t0 tf]*, and an initial condition, which are the same as the position matrix defined in the paragraph about task 3. I used the math from slides in the powerpoint to implement euler's method. The function is to create our own *odeXX*. The function *euler* is used to solve differential equations. I tested this function by giving it the same conditions as the *ode45* in the *test\_spaceship\_motion.m* file and also plotted my function against the *ode45* and I got nothing similar to an ellipse. I got a straight line above the ellipse in the graph. I am not sure if I did this task correctly.

For the sixth task I implemented the Runge-Kutta method and the Ralston method. I copied my code from the *euler* function and changed the equations to match the specific method. I tested these functions as I tested the *euler* function and I received similar results.

I did not attempt task seven because I didn't quite understand how to attack this task and due to the amount of time I also have to spend on my other subjects.