Computational Physics

Exam 01

1. a. A Python list has square brackets and can contain a variety of different objects at the same time separated by commas, while a tuple has parenthesis with objects separated by commas and the objects in the tuple cannot be changed, reassigned, or modified. Finally, an array is the most useful and it contains numbers in a grid, like a matrix, which can be modified by mathematical operations.

b. The difference between a for and a while loop is in the name of the loops themselves. A for loop will continue loop lines of code ‘for’ a certain known range or number of loops. A while loop will continue looping ‘while’ a certain condition is true and you is generally used when you do not know the number of iterations of the loops are necessary. The for loops is the better loop for not getting stuck in an infinite loop because it has a set number of loops but the while loop will continue as the condition is true and it is always true it will not stop, hence you will be stuck in an infinite loop.

c. Everything in Python is an object meaning that it has data and functions as part of it. That means everything, even something as simple as numbers, have ‘rules’ defined to them. Like the difference in the object ‘integer’ and the object ‘float’ which may both seem like numbers but have different data and functions associated with them. This can be helpful because python is doing some of the work for you as a background process but at the same time you could lose track of what your object is or what is does resulting in errors in your code.

d. Object: Windsor Chair:



Attributes: This object exist within a 3-dimensional space plus time and therefore can be represented by an array with numerical values for its size and location while a value for the time is varying (increasing usually). It has some mass which describes how it acts inside a gravitational field. It is made of a material (let’s just say ‘wood’) that allows it to be sturdy, lasting, burnable, and other attributes that the object wood possesses. Lastly it has style, which sets it apart from how other chairs may be in color, age, and craft.

Methods: The chairs primary function is ‘sitting’ which is when the object ‘human’ interacts with the chair is such as way that a human puts their mass on the chair, and both remain in a state of equilibrium. Another function of the Windsor chair is ‘holding’ which is similar to sitting except it is when the chair supports the mass of any non-living object for an indefinite amount of time. Lastly, the chair has the function decoration which gives an attribute to the object room of organization or style.

1. To perform this comparison on whether a to b or b to a is more accurate I used my script for exercise\_09 and used an h = 0.001, an array of x values from 0 to 10\*\*4 and an array of x values from 10\*\*4 to 0, while making h = -0.001. This was done with the function x^2. Knowing that the actual integral computed by hand should have nothing but three’s, I compared the two to see which one deviated first. The computational limit definition of the integral from 0 to 10\*\*4 was 333333183333.0 while the value from 10\*\*4 to 0 was 333333383265.0. We could look at the relative error between these two and the actual value of 333333333333.3 … but because there are so many digits the error will be small and it is easy to tell that because the first value 333333183333.0 deviates first, that is to say the 1 in the hundred thousands place comes before the 8 in the ten thousands place for the second number, the integral evaluated from 0 to 10\*\*4 is less accurate than evaluated 10\*\*4 to 0. We looked at a similar problem in exercise\_05 in which we summed up and we summed down and noticed that summing down worked much better. The reason for this is because Python only keeps up to 16 digits and if you sum down then the first digit is represented in scientific notation rather than including all 16 digits but when you sum up once you get to 16 digits and you must keep summing it starts dropping digits and changes to scientific notation within the summation and loses decimals along the way. We talked about this in class in exercise\_05 and how summing down is better than summing up. Therefore, summing from 10\*\*4 to 0 is more accurate than summing from 0 to 10\*\*4.

This was tested again for an h = 0.0001 and the values were High to Low 333333340691.0 and Low to High 333333318333.0 checking for the percent error again with these numbers I get that High to Low yields 2.2E-6 percent and Low to High yields 4.5E-6 percent. Again, showing that summing High to Low is more accurate

1. The generalized equation for a system of variable mass and velocity, which would describe a rocket is F\_ext = **m**(d**v**/dt) – (**u** - **v**) \* d**m**/dt. F\_ext is the external forces for the rocket which would include gravity and air resistance, **m** is the mass of the rocket that changes with time d**m**/dt as its derivative, **v** is velocity, in x, y, and z space and d**v**/dt is the acceleration, and lastly **u** is the velocity of the particles emitted from the rocket. Let’s call **u – v =** **V** which will be the relative velocity. If we assume the air resistance is linear, which makes our problem simpler, we still seem to have a complicated differential equation. We need to find the velocity as a function of time and the amount of fuel it takes and its cost to put the rocket into geostationary orbit. Let’s ignore the x and y variable of our velocity and assume it travels straight up, and that the force due to air resistance is F = -k**v** where **v =** d**z**/dt. Let’s also assume that **V** is constant(V), that is to say the relative speed between the rocket and the particles emitted stays the same. And force due to gravity is known, so we are left with:

g– k(d**z**/dt) = **m**(d**z**/dt) – V(d**m**/dt)

This differential equation can be integrated and solved using standard numerical and possibly even analytical practices. As was mentioned in class Monte Carlo Integration methods are used when you have many dimensions (or anything that has a long computation time), like having ten dimensions, so that the computation time is not millions of years. In this case, even though it has been simplified, we still would not need a Monte Carlo method of integration to solve for this equation.