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 ‘adding’ which is when the chair object adds a human object so that the chair supports the mass of the human. Another function of the Windsor chair is ‘holding’ which is similar to adding except it is when the chair adds on the mass of any non-living object for an indefinite amount of time. Lastly, the chair has the function ‘scuffing’ which creates cuts or marks on the floor object.

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 thousand 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 and looking at the difference between these and the actual value for the integral we see that:

Comparison of the difference between low-high to high low

Low to High gives: -7357.44030762

High to Low gives: 14999.9397583

The Low to High integrations gives a smaller difference in magnitude showing that it is the more accurate method.

Forethought: The results can vary based on parameters of h, whether or not you go to the full range ((10^4)^2) or one value less than it, and how the code was written. Logically, it makes more sense however to start with really small numbers that are in scientific notation and add those first because if you start with really large numbers then when you add on the smaller it makes little difference and the number could be lost but starting small you add up to larger numbers and less of your number is ‘chopped off.’

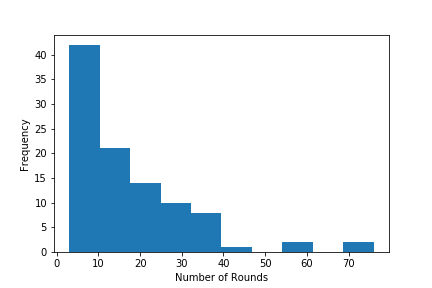
1. The generalized equation for a system of variable mass and velocity, which would describe a rocket is F\_ext = **m**(d**v**/dt) – (**u** - **v**) \* d**m**/dt. F\_ext is the external forces for the rocket which would include gravity and air resistance, **m** is the mass of the rocket that changes 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)

Now this equation has the constants g, k, and V none of which are random or changing, hence being constants. Our variables are the vertical displacement **z**, the mass **m**, and the change in mass d**m**/dt. This results in at two dimensional integral. Now Monte Carlo Methods of integration can be used to save time and effort for higher dimension integrals like that of calculating the properties of Magnesium and its 12 electrons which results in finding the wave equation, with its x, y, and z components for 12 different electrons which is a 36 dimensional integral. Solving this with normal computational methods of integration would take a time longer than the age of the universe but by employing randomness to find randomly calculate for certain values in the integral rather than all the values will be significantly faster. And we know by the nature of randomness our result will be quite accurate due to statistical spread. However, in the case of this rocket with a two dimensional integral and that while we might want to know a specific value of the fuel spent to find cost we also want our vertical displacement **z** as a vector equation to give us a trajectory. Because of this, using a Monte Carlo Method, which is to say randomly choosing points in the integration and summing them to find our value, is not necessary and not needed for this particular problem.

1. We can see that the older version of High-Ho-Cherry-O that has even spacing has more games that can take more rounds. The newer version, which has twice the size spacing for each section in which you can pick a cherry, never has a game lasting more than about 35 rounds. There is also a higher frequency of games lasting about 5 rounds for the uneven spaced version. The uneven spaced version would be much better for children because kids have a short attention span and honestly nobody wants to play for 70+ rounds.

Histogram of even spaced game:



Histogram of uneven spaced game:

