**READ ME: electronbeam.py**

The main method is set up to simulate an even fluence over the field size. See the comments at the start for variable definitions, and define them as required.

**Description of Classes and Functions (top to bottom in the program):**

**Methods:**

**Main Method:** this is called to run the program. It creates an electron object, and it evenly starts 1 million electrons across a 4x4cm2 field and transports them into a phantom object until they are out of energy, and all dose has been deposited.

**Electron\_paths:** this method can be run to keep track of electron positions while they are propagated. I used it to create the electron paths figure on the title page of my report.

**CSV2Array:** this method is used to take a CSV containing electron stopping power data, and convert it into an interpolation function, which inputs an energy and outputs a stopping power.

**closestIndex:** this method takes in an array and a value, and find which array index corresponds to a number closest to value. This is used for determining which voxel in the phantom an electron is currently in.

**phiScatterAngle:** randomly samples the azimuthal scatter angle

**thetaScatterAngle:** Uses the multiple scattering theory to generate an electron scatter angle based on energy and step size.

**plotCAXdose:** Method is called on to create the electron PDD figure, after all electrons have been simulated.

**CSDA\_particle:**  This plots the analytical CSDA electron curve when there is no multiple scattering.

**Classes:**

**Electron:** create an instance of an electron with a starting position, direction, and call its transport() method to propagate it until its out of energy.

**Phantom:** create a phantom object with a certain size, which will be used to hold the dose deposited by electrons.

**Paths:** A paths object is created to hold the path of electrons when a paths figure is created.