Primer Reporte

El objetivo de la primer entrega es simular la representación de mallas de átomos con patrones cuadrados o hexagonales con una periodicidad dada con el fin de observar la nueva malla formada al colocar una sobre otra, para esto usamos el lenguaje Python auxiliado con Jupyter Notebook.

Se presentan varios métodos que usamos para cumplir nuestro propósito, para imprimir en pantalla (y en una imagen png si se requiere). Tenemos el método *printPoints2(pList1,pList2,r)*, el cual dibuja en una sola imagen discos de radio r con centro en las posiciones dadas por las listas pList1 y pList2 respectivamente. Para generar una lista con la posición de puntos con simetría cuadrada usamos el método *rectangles(nX,nY,pX,pY)* donde nX y nY son el numero de veces que repetiremos en las direcciones 'x' y 'y', pX y pY la distancia entre ellos. De igual forma el método *hexagonal(nX,nY,p)* genera una lista de posiciones pero con estructura hexagonal que se repite nX veces en la dirección x y nY veces en sus dirección en ángulo con un periodo P.

Pese a cumplir el objetivo, se vieron varias deficiencias, la primera es en el tiempo que se toma en graficar la maya, otro problema está en el hecho de que las Mallas creadas se limitan a ser sólo rectangulares (sus ángulos internos son rectos) o hexagonales regulares, además no de no poderse rotar.

Segundo Reporte

Los objetivos para esta entrega son 2 principalmente, el primero es crear una nueva representación de mallas usando sus Vectores constructores u y v pudiendo hacer que con estas mallas se puedan representar ‘átomos’ ordenados con patrones tanto cuadrados cómo hexagonales, pudiendo también hacer que el color de los ‘átomos’ en cada celda de la malla puedan ser distintos. El segundo objetivo es poder rotar en un ángulo dado la maya.

En el nuevo código tenemos 2 clases, *‘Atomo’* que será la representación de los átomos contenidos en la malla y la clase *‘Malla’* que es la representación de la malla en si. La construcción de un objeto *‘Atomo’* requiere de entrada su posición (x,y) y opcionalmente el color con el que será dibujado, un Objeto Maya requiere por entrada 2 vectores constructores 'u' y 'v' y opcionalmente una variable 'Theta' que indicará el ángulo de rotación de la maya si es que esta está rotada, si no se especifica se le dará un valor 0 por default.

Para crear los átomos en una Malla usamos el en dicha Malla el método ‘crea(n,m,nAtms,colors)’ donde n y m son el número de veces que los vectores u y v de la malla respectivamente en sus 2 direcciones cada uno, nAtms el número de átomos dentro de cada celda de la malla (si es 1 se crean mallas con átomos en patrones cuadrados y si es 2 tendrán patrones hexagonales) y opcionalmente colors será una lista con los colores de los átomos en las celdas de la malla, si no se indica serán todos azules. Para ver en pantalla usamos el método *‘muestra’* que recibe una lista de Mallas y las imprime juntas en pantalla.

Se soluciona el problema del tiempo de la impresión de la malla en pantalla (y en un documento de imagen si así se indica) con un cambio con respecto a la anterior usando la función scatter de matpltlib en lugar de patch reduciendo el tiempo de dibujo pero generando problemas en la representación cuando la diferencia del tamaño entre los Átomos con la Maya completa es muy grande.