***Panighel Fernando***



*DNI: 37.983.056*

*Taller de Matemática Computacional*

*Primer Año*

*Facultad de Ciencias Exactas, UNCPBA*

**Introducción:**

Mediante Octave, nuestra nave espacial nos permite hacer un pedido de autorización para disparar usando nuestro DNI en base a la función dada ***my\_mex\_service.***

***Desarrollo:***

Use método de probabilidades de Montecarlo. Creé la función ***calcular\_probabilidad\_realizar\_disparo.*** La misma toma mi **DNI** + 3 diferentes valores de ***épsilon*** para calcular sus diferentes resultados. Dentro de esta función trabajamos con:

1- **Probabilidad\_anterior**: Probabilidad que se compara con la actual para la convergencia.

2- **Probabilidad\_actual**: La que se compara con la probabilidad anterior.

3- **Casos\_favorables**: se incrementa si no se autoriza el disparo dos veces consecutivas.

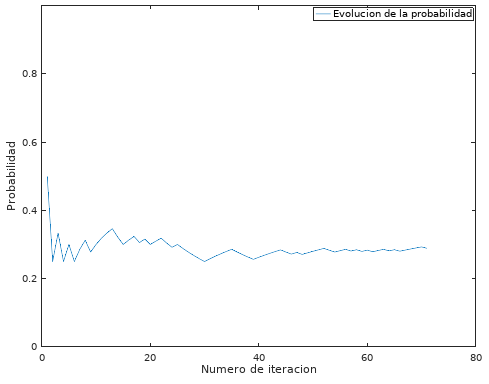
4- **Total\_pedidos**: Todos los pedidos de disparos que hago.

5- **Probabilidades\_parciales**: arreglo contenedor de iteraciones.

6- **Resultado (1 y 2)**: variable donde se almacenaba un 1 en caso de disparo autorizado o un 0 en caso de devolución negativa. Ambas variables se compararían para sumar o no un caso favorable.

Hice un bucle que se termina cuando la función converge. La función “converge” recibe como parámetro *la* *probabilidad anterior, la actual, el total de pedidos hechos y un valor de épsilon* luego, compara el valor absoluto de la resta de *la probabilidad anterior* con la *probabilidad actual* y si es menor a épsilon y el total de pedidos era mayor al número dado devuelve un valor booleano true, sino false. Dentro de este bucle principal (while converge) la *probabilidad anterior* se volvía la *actual*, se hacían los dos pedidos de autorización a la nave, se incrementaba en dos unidades el total de pedidos realizados y se los comparaba, si ambos eran negativos se incrementaban los casos favorables una unidad. Finalmente la probabilidad actual se calculaba como el cociente de la división entre los casos favorables con la cantidad de pedidos y se guardaba el valor concatenándose en el arreglo de las probabilidades parciales. Estos dos resultados eran lo que la función “*calcular\_probabilidad\_realizar\_disparo”* devolvía.

***Resultados:***

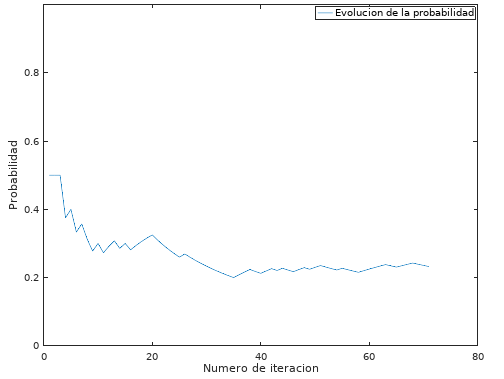
***Épsilon:*** *0.1*

***Probabilidad:*** *0.288732*

***Primeras 20 iteraciones:*** *0.052167*

***Últimas 20 iteraciones:*** *0.003950*

***Tiempo de cálculo:*** *0.942054*

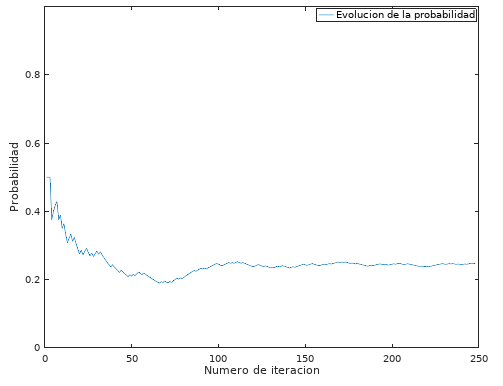
***Épsilon:*** *0.01*

***Probabilidad:*** *0.232394*

***Primeras 20 iteraciones:*** *0.075441*

***Últimas 20 iteraciones:*** *0.007446*

***Tiempo de cálculo:*** *0.952054*

***Épsilon:*** *0.001*

***Probabilidad:*** *0.245968*

***Primeras 20 iteraciones:*** *0.069628*

***Últimas 20 iteraciones:*** *0.001026*

***Tiempo de cálculo:*** *1.094063*

***Conclusiones:***

A menor épsilon, mayor el tiempo de cálculo y más precisa la probabilidad. Luego las iteraciones iniciales y finales nos ayudan a declarar que el experimento toma su forma cuanto mayor es la cantidad de disparos, equilibrando la probabilidad. Las ultimas iteraciones tienen una variación diferente según el épsilon dado, cuanto menor es el épsilon menor es desvió en las ultimas iteraciones.