



*UNIVERSIDAD NACIONAL DEL CENTRO
DE LA PROVINCIA DE BUENOS AIRES*

Taller de Matemática Computacional

2017

**Alumno: Castelli, Emiliano
DNI: 34336822**

Introducción:

El trabajo especial consistía en resolver un problema con la herramienta Octave. El problema intentaba reunir todo lo que se estuvo viendo durante el cuatrimestre en un único ejercicio para simular la probabilidad de que un disparo de la nave espacial falle dos veces seguidas.

Desarrollo:

La implementación realizada consiste en archivos de Octave creados desde cero y otros reutilizando código de ejercicios realizados anteriormente.

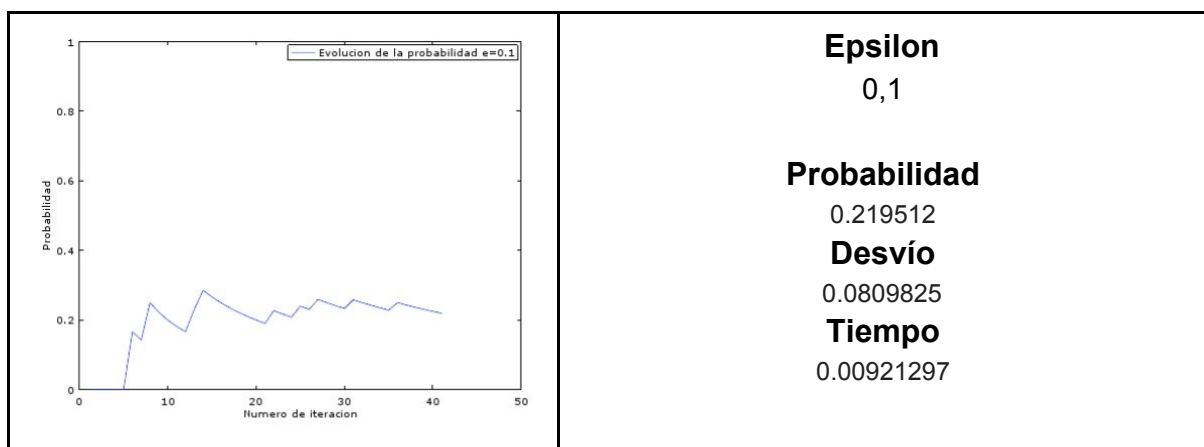
Se reutilizó el script inicializar_o_restaurar.m modificando el path que se incluye en el archivo para poder utilizar las funciones adicionales.

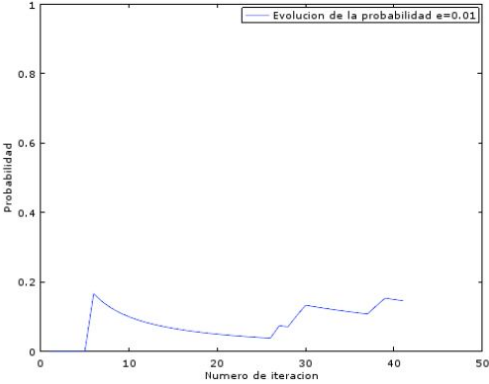
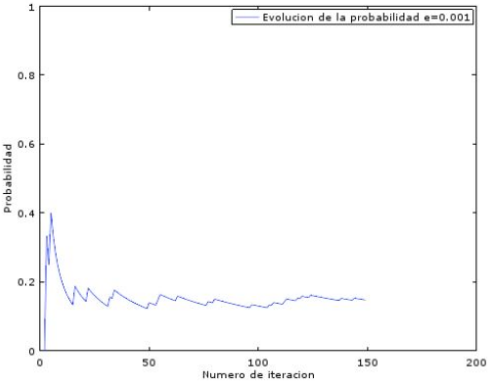
Se creó el archivo prob_2_fallos_sucesivos.m, en base a un ejercicio anterior que calculaba probabilidad de que salga cierto lado de una moneda, que resuelve el inciso 1 del trabajo práctico especial (calcula la probabilidad de que ocurran dos fallos sucesivos dado un número de DNI y un epsilon).

Se creó el archivo script_trabajo_especial.m, con el fin de que realice todos los incisos que se plantearon en el trabajo. Este archivo fue sufriendo modificaciones debido a que cada vez se le incorporaron más funcionalidades.

Debido a reiterados fallos de la función my_mex_service, se programó primero una función para suplantarla (my_mex.m) que devolvía cero o uno en base a una probabilidad aleatoria. Luego se perfeccionó para que funcione de forma similar a la propuesta por el código my_mex_service.c para que en el cálculo de la probabilidad intervenga también el DNI del alumno.

Resultados:



	<p>Epsilon 0,01</p> <p>Probabilidad 0.146341</p> <p>Desvío 0.0476743</p> <p>Tiempo 0.008003</p>
	<p>Epsilon 0,001</p> <p>Probabilidad 0.147651</p> <p>Desvío 0.0725525</p> <p>Tiempo 0.026017</p>

Conclusiones:

En el ejemplo mostrado, y en tantas otras simulaciones, siempre el valor de ϵ influyó en el resultado de la probabilidad. Un número de ϵ más pequeño, nos daba una probabilidad más exacta.

Como se puede apreciar, con $\epsilon=0.1$ el tiempo de ejecución fue menor que con $\epsilon=0.001$ ya que se realizaron menos iteraciones hasta converger. Cuando $\epsilon=0.01$ el tiempo de ejecución es similar debido a que se realizan aproximadamente 40 iteraciones.

Lo curioso es que para $\epsilon=0,01$ se obtuvo una probabilidad similar a cuando $\epsilon=0,001$ y en mucho menor tiempo aunque con datos más dispersos como se puede ver en el parámetro del desvío calculado.

Como conclusión podemos decir que cuanto más cercano a cero elijamos el epsilon, más cerca estaremos de la probabilidad teórica aunque a costa de mayor tiempo de ejecución del programa realizado.