implementacion de contrastes de uniformidad y aleatoriedad

***Abstract*— The work consists of several stages from programming random number generators this through various methods, later development Implementation contrast uniformity and randomness by different test**

Resumen - El trabajo realizado consta de varias etapas desde la programacin de generadores de nmeros aleatorios esto a travs de varios mtodos, Posteriormente el desarrollo Implementacin De contraste de uniformidad y aleatoriedad por medio de diferentes test

I. INTRODUCTION

Los seres humanos vivimos en un medio aleatorio y nuestro comportamiento lo es tambin. Si deseamos predecir el comportamiento de un material, de un fenmeno climatolgico o de un grupo humano podemos hacerlo a partir de datos estadsticos, pero para lograr una mejor aproximacin a la realidad nuestra herramienta predictiva debe funcionar de manera similar, es decir, aleatoriamente.

Nmeros aleatorios: Son obtenidos al azar es decir, son resultado de un proceso en el cual su resultado no es predecible ya que todo numero tiene la mimas probabilidad de ser elegido y la eleccin de uno no depende de la eleccin del otro.

La importancia de los nmeros aleatorios radica en su uso para la generacin de variables aleatorias que son requeridas en los experimentos de simulacin.

II. METODOLOGIA

La metodologa es sencilla se usaron tres pasos: 1. la codificacin de los algoritmos para generar variables aleatorias. 2. codificacin de los algoritmos de contrastes de uniformidad y aleatoriedad. 3. puesta a prueba de los algoritmos de generacin de variables con los test de uniformidad y aleatoriedad.

A. *Maintaining the Integrity of the Specifications*

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations

III. COMO GENERAR NUMEROS ALEATORIOS?

con pocas lineas de codigo en cualquier lenguaje de programacion se pueden generar numeros aleatorios. Estos algoritmos producen una sucesion de numeros que se asemeja a la de una sucesion de realizaciones de variables aleatorias iid U(0, 1)

A. *Metodo congruencial*

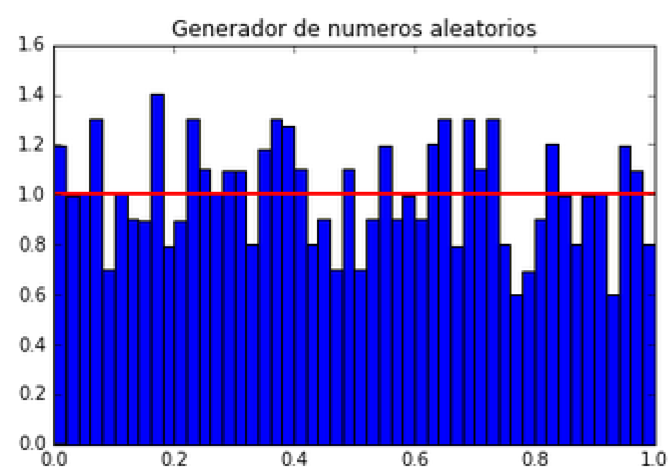
Un metodo congruencial comienza con un valor inicial (semilla) x_0, y los sucesivos valores x_n, n 1 se obtienen recursivamente con la siguiente formula:

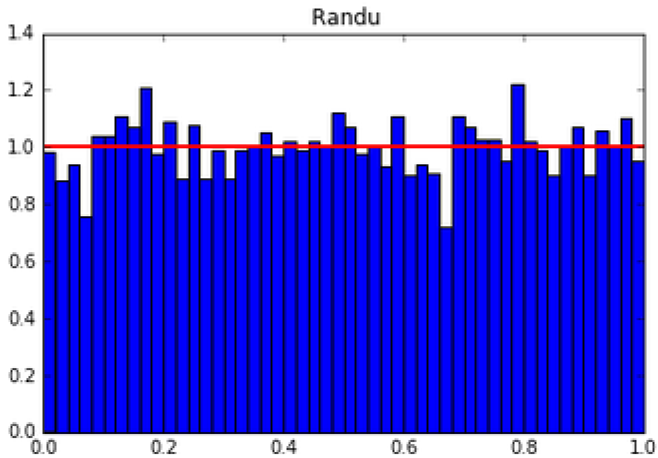
$$x_n = a(x_n - 1) + b(modulom)$$

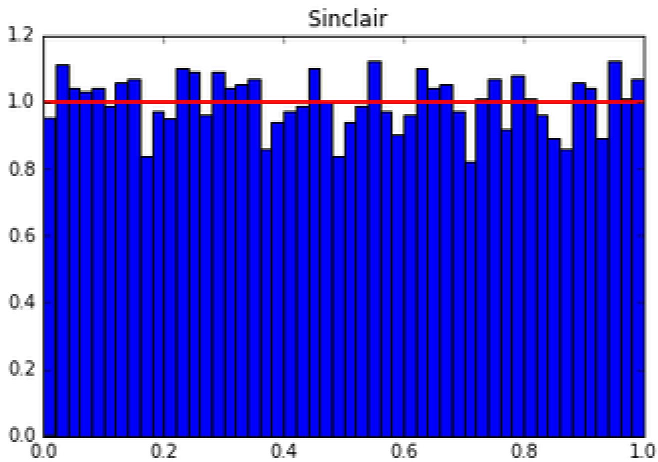
existen diferentes generadores congruenciales ya definidos como:

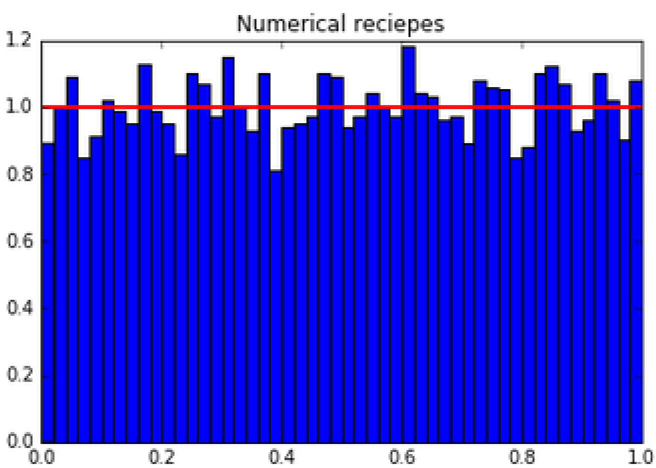
- RANDU $x_{i+1} = 65539 x_i \mod 2^3 1$
- Sinclair ZX81 $x_{i+1} = 75 x_i \mod (2^{16} + 1)$
- Numerical reciepes $x_{i+1} = 1664525 x_i + 1013904223 \quad m$
- Borland C/C++ $x_{i+1} = 22695477 x_i + 1 \mod 2^{32}$

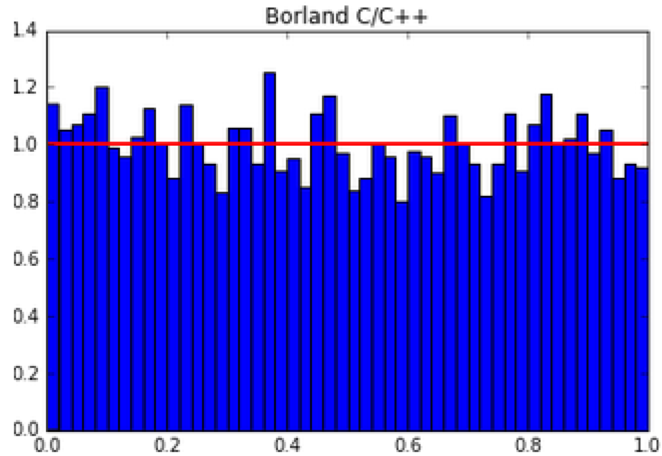
graficando en python los generadores se tiene como resultado











a simple vista se puede ver que los generadores no estn lejos de ser totalmente uniformes para confirmar esto se realizan la pruebas de uniformidad

B. Pruebas de uniformidad

Se trata de decidir si los numeros generados se pueden considerar como una realizacion de una muestra aleatoria simple de una distribucion U(0, 1). entre estos estan:

- Constraste o prueba de Kolmogorov-Smirnov
- Contraste o prueba χ^2

en el desarrollo de la actividad de implementara el metodo de prueba de Kolmogorov-Smirnov

C. Kolmogorov-Smirnov

El test de K-S es un test de bondad de ajuste que se utiliza para determinar si los datos de una determinada muestra se ajustan a una hipotetica distribucion.

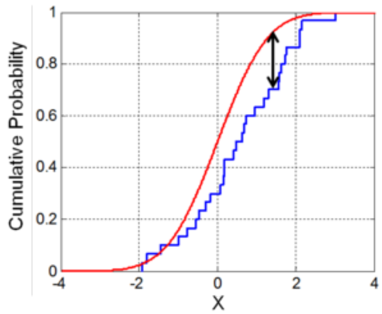
Consideramos el caso en que F_0 es continua. La funcin de distribucin emprica de una muestra $X_1, X_2, ..., X_n$ se define como:

$$F(x) = \frac{\#\{X_i \leq x\}}{n}$$

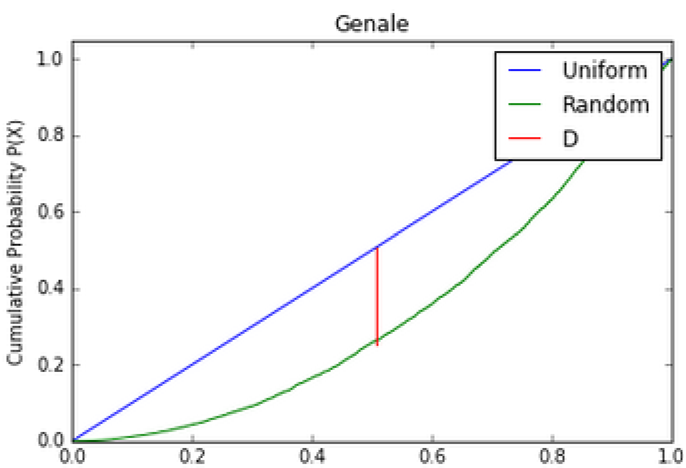
Bajo la hiptesis nula $H_0 : F_X(x) = F_0(x)$, esperamos que F_n se aproxime a F_0. Definimos el estadstico bilateral de Kolmogorov-Smirnov

La distribucin exacta de D_n est tabulada para valores seleccionados de $n \leq 40$ y del nivel de significacin α. Para muestras grandes, se utiliza la distribucin asinttica de D_n, que viene dada, para todo $z \leq 0$, por

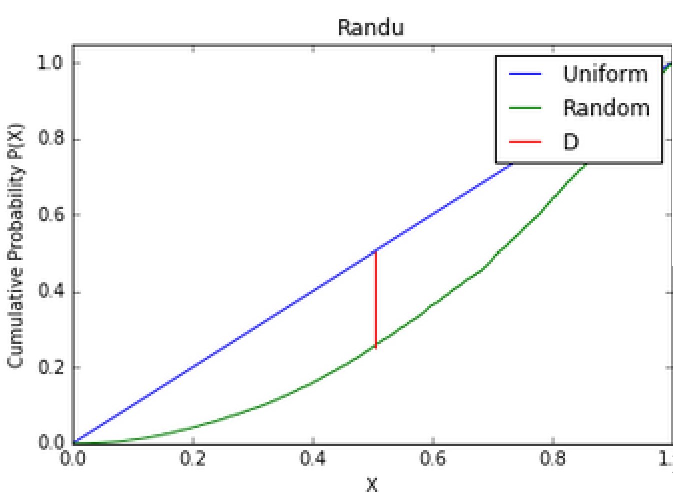
$L(z)$ est tabulada y se comprueba que la aproximacin es suficientemente buena para $n \geq 35$. Intuitivamente, esperamos que D_n sea pequeo cuando la hiptesis nula es cierta. En nuestro caso particula de aleatoriedad, si $X_{(1)} < X_{(2)} < ... < X_{(n)}$ designa al estadstico de orden, $F_0(x_{(i)}) = x_{(i)}$, y como $F_n(x_{(i)}) = i/n$, resulta:



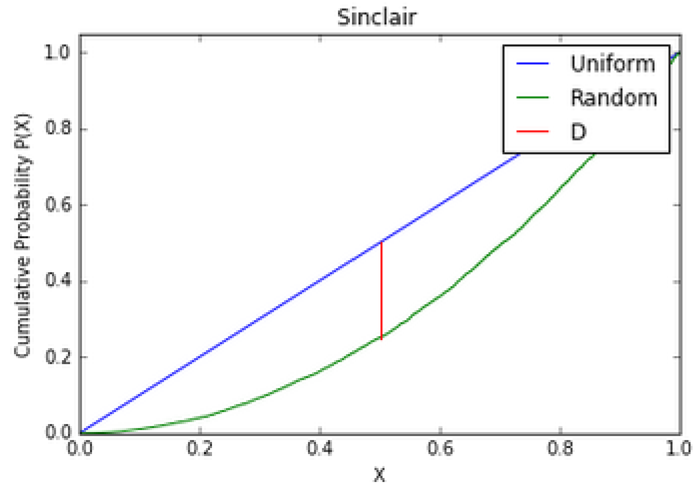
D. se prueban los generadores por el metodoKolmogorov



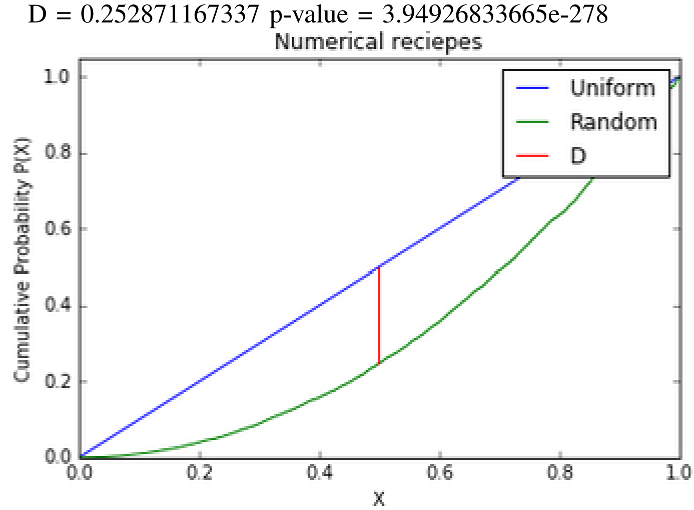
D = 0.25711097451 p-value = 1.60726943315e-287



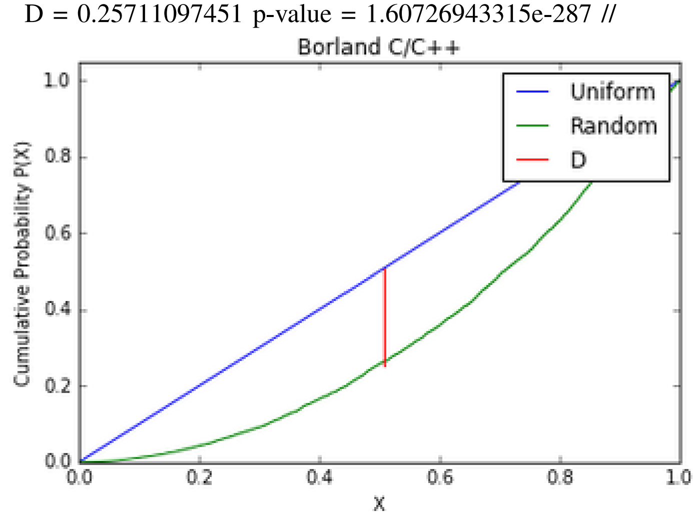
D = 0.248256113801 p-value = 4.37091211326e-268



D = 0.252871167337 p-value = 3.94926833665e-278



D = 0.25711097451 p-value = 1.60726943315e-287 //



D = 0.25711097451 p-value = 1.60726943315e-287

IV. Contraste de aleatoriedad

permite verificar la hiptesis nula de que la muestra es aleatoria, es decir, si las sucesivas observaciones son independientes.

En algunos casos, los datos muestran una patrn claramente no aleatorio (por ejemplo si una variable debe presentar valores aleatorios que sean enteros entre 0 y 9, la secuencia "4 3 2 1 0 4 3 2 1..." es poco probable ya que en ningn caso los valores exceden el valor 4). Si un conjunto de datos no pasa la prueba de aleatoriedad, entonces puede ser sustituida por otra serie de datos aleatorizados que pase el test de aleatoriedad.

entre estos estan:

- Test de rachas
- Contraste de rachas por encima y por debajo de la mediana
- Contraste o prueba de permutaciones
- Contraste o prueba de huecos

en este caso se uso el test de rachas que sera el que se muestre a continuacion

A. *Constraste o prueba de rachas*

Este contraste se basa en el nmero de rachas que presenta una muestra. Una racha se define como una secuencia de valores muestrales con una caracterstica comn precedida y seguida por valores que no presentan esa caracterstica. As, se considera una racha la secuencia de k valores consecutivos superiores o iguales a la media muestral (o a la mediana o a la moda, o a cualquier otro valor de corte) siempre que estn precedidos y seguidos por valores inferiores a la media muestral (o a la mediana o a la moda, o a cualquier otro valor de corte).

Dada la sucesin de observaciones $X_1, X_2, ..., X_n$, construimos la sucesin de simbolos binarios definida mediante 1 si $X_i < X_{i+1}$, 0 si $X_i > X_{i+1}$. Definimos racha creciente (decreciente) de longitud l a un grupo seguido de l nmeros 1 (0). Contabilizamos el nmero de rachas. Sabemos que su distribucin asinttica, bajo la hiptesis nula de aleatoriedad, es:

$$N\left(\frac{2n-1}{3}, \frac{16n-29}{90}\right)$$

Intuitivamente, rechazamos la aleatoriedad con un nmero muy pequeo o muy grande de rachas. De ah se obtiene inmediatamente la prueba.

B. *probando los generadores anteriores con el test de rachas*

generador de numeros aleatorios propio
test de rachas con 5000 datos
numero de rachas de la lista = 3287
'Z = -1.54316662878'

RANDU
test de rachas con 5000 datos
numero de rachas de la lista = 3327
'Z = -0.201282603754'
Sinclair
test de rachas con 5000 datos
numero de rachas de la lista = 3298
'Z = -1.1741485219'

Numerical reciepes
test de rachas con 5000 datos numero de rachas de la lista = 3317
'Z = -0.53675361001'
Borland C/C++
test de rachas con 5000 datos
numero de rachas de la lista = 3356
'Z = 0.771583314389'

V. CONCLUCIONES

para los contrastes de uniformidad podemos ver a simple vista que generadores no tan salidos de la uniformidad pero as mismo con las pruebas de Kolmogorof y sus resultados al ser nmeros tan pequeos podemos concluir que los generadores son validos para una distribucin uniforme

en las pruebas de aleatoriedad los generadores consiguieron los resultados desde -1.5431 hasta 0.771583 con lo que se puede concluir que tienen una aleatoriedad aceptable.

con los resultados obtenidos tambin podemos concluir que el generador RANDU es el menos aceptable de los generadores probados y que para mejores nmeros aleatorios es conveniente usar nmeros primos, grandes e impares.

REFERENCES

[1] - Simulacin y anlisis de sistemas con ProModel Garcia Dunna Eduardo, Garcia Reyes Heriberto. Cardenas Barron Leopoldo E
[2] - Simulacin un enfoque prctico. Ral Coss Bu. Editorial Limusa.