

Números pseudoaleatorios

Física de los Sistemas Complejos
4º Física, UGR

Febrero 2018

Número aleatorio

- ▶ Resultado de una variable al azar proveniente de una función de distribución.

Distribución uniforme $\rightarrow U(a, b)$

Distribución normal $\rightarrow N(\mu, \sigma)$

- ▶ ¿Por qué son necesarios?
 - ▶ Simulaciones
 - ▶ Análisis numérico
 - ▶ Criptografía
- ▶ Generación de número (pseudo)aleatorios (secuencias sin significado físico). Un ordenador es una máquina determinista.

Generador congruencial

Ejemplo

Generador de enteros dado por

$$X_{n+1} = aX_n + c \pmod{m}$$

donde a y c son constantes, que en este caso consideraremos $a = 5$, $c = 1$ y $m = 16$, con semilla $X_0 = 1$.

Ejercicio

Genera una serie de números aleatorios congruenciales dada por

$$X_{n+1} = \text{mod}(cX_n + a_0, N_{\max})$$

con parámetros, $c = 65539$, $a_0 = 0$, $x_0 = 1$ y $N_{\max} = 2^{31} - 1$.

Localiza un error al generarlos. ($\frac{X_n}{N_{\max}} \Rightarrow U(0,1)$).

Distribuciones de probabilidad

- ▶ Se pueden generar secuencias de enteros y distribuciones uniformes.
- ▶ ¿Cómo se genera una distribución de probabilidad?
Distribución acumulada

$$F(x) = u = U(0, 1)$$

- ▶ Ejemplo: Distribución gaussiana

Fortran/C

- Funciones rand, random number, random



Random numbers repeat with different recurrent periods.

```
rand:
1st number is repeated at 1511326 th number
2nd number is repeated at 265037 th number

random_number:
1st number is repeated at 481954 th number
2nd number is repeated at 406187 th number

random:
1st number is repeated at 134457 th number
2nd number is repeated at 134458 th number
```

All random numbers repeat with a same recurrent period.



Gnu Scientific Library



GNU Operating System

Sponsored by the Free Software Foundation

[ABOUT GNU](#) [PHILOSOPHY](#) [LICENSES](#) [EDUCATION](#) [SOFTWARE](#)

GSL - GNU Scientific Library

<https://www.gnu.org/software/gsl/>

FGSL: A Fortran interface to the GNU Scientific Library

This web page provides the source code for a portable, object-based Fortran interface to the GNU of numerical routines for scientific computing. The package is licensed under the GPL. Please be aware that the package is at your own risk.

[https://www.lrz.de/services/software/mathematik/gsl/
fortran/](https://www.lrz.de/services/software/mathematik/gsl/fortran/)

Generación de aleatorios

Fortran/C

```
#include <stdio.h>
#include <gsl/gsl_rng.h>

int
main (void)
{
    int i, n = 10;
    const gsl_rng_type * T;
    gsl_rng * r;

    gsl_rng_env_setup(); //Inicializacion de las gsl
    gsl_rng_default_seed=0; //Permite cambiar la semilla por defecto

    T = gsl_rng_mt19937; //gsl_rng_default; // Generador a utilizar
    r = gsl_rng_alloc (T); //Comprueba si hay suficiente memoria para emplear el generador
    gsl_rng_set(r,35); // Otra forma de cambiar la semilla, pero despues de llamar al generador. Elegir este caso o el anterior

    for (i = 0; i < n; i++)
    {
        double u = gsl_rng_uniform (r);
        unsigned long int h = gsl_rng_uniform_int (r,100);
        printf ("Uniforme [0-1): %.5f          Entero entre 1 y 100: %lu \n", u, h);
    }

    gsl_rng_free (r); //Libera toda la memoria que haya empleado el generador

    return 0;
}
```

```
1 program random
2 use fgsl
3
4 implicit none
5
6 integer*8 :: seed
7 type(fgsl_rng) :: r
8 type(fgsl_rng_type) :: ini_fgsl
9
10 !Inicializo aleatorios con las gsl.
11 seed=21
12 ini_fgsl = fgsl_rng_env_setup()
13 fgsl_rng_default_seed=seed
14 ini_fgsl = fgsl_rng_default
15 r = fgsl_rng_alloc(ini_fgsl)
16
17 write(*,*) fgsl_rng_uniform(r)
18
19 stop
20 end
```


Caminante aleatorio

Random walk

Considerad el siguiente proceso estocástico

$$x(t+1) = x(t) + \eta(t)$$

donde $\eta(t)$ es un ruido blanco gaussiano con $\langle \eta_i(t) \eta_j(t') \rangle = \delta(t - t') \delta_{ij}$. Generando muchos caminantes, representad la posición media del caminante aleatorio y la evolución de su varianza en función del tiempo.