

# **Simulación a Eventos Discretos**

Tema 4: Modelado estocástico (parte 2)

# Muestreos

- Los problemas que modelamos en general son estocásticos.
- No podemos predecir la conducta de los elementos del sistema, pero sí podemos enumerar los resultados posibles de ella.
- Entonces utilizamos técnicas de *muestreo de distribuciones*.
  - Generación de *números aleatorios*: se utilizan para obtener muestras independientes de variables aleatorias.
  - Muestreo de *variables aleatorias*: se obtienen a partir de números aleatorios.

# Muestreos

- Para simular el comportamiento del sistema utilizamos muestras de variables aleatorias.
- Si queremos observar el comportamiento general del sistema alcanza con realizar una sola corrida (replicación o ejecución de la simulación).
- Si queremos obtener medidas o valores estadísticamente válidos deben realizarse varias corridas con distintos números aleatorios que generarán distintas muestras de las variables aleatorias.

## Generación de números aleatorios

- Rol fundamental en el proceso de simulación cuando el aspecto estocástico de sistema es relevante.
- Para simular necesitamos de números aleatorios para generar muestras de variables aleatorias.
- Características deseables de un generador de números aleatorios:
  1. Genera valores *uniformemente distribuidos* en un intervalo dado.
  2. Asegura la *no correlación serial*.
  3. Otras, ver textos de Law (2015) y Banks et al. (2009) de la bibliografía del curso.

## Propiedades de los números aleatorios

**Distribución uniforme** Cualquier número que pertenezca al rango de interés debe tener la misma probabilidad de resultar sorteado.

**No correlación serial** La aparición de un número en la secuencia, no afecta la probabilidad de sortear otro (o el mismo) número.

Ejemplo: la sucesión 1,2,3,4,5,1,2,3,4,5... podríamos decir que es uniforme pero está correlacionada.

Existen tests que verifican las propiedades de uniformidad y correlación serial.

## Números pseudoaleatorios

Imitan los valores de una variable aleatoria uniforme. Cumplen los tests de ajustes como si fueran esa variable aleatoria.

Se generan a través de una fórmula y se usan como base para generar valores de variables aleatorias (discretas, continuas).

Son *pseudoaleatorios* porque se obtienen realizando un conjunto de operaciones a partir del número generado en algún paso anterior.

Ventaja: método muy veloz y barato, permite reproducir experimentos.

Desventaja: son de período finito (el período es la cantidad de números diferentes de la secuencia generada, hasta que se genera un número ya generado).

## Números pseudoaleatorios

Tanto las secuencias como las subsecuencias de los números generados deben cumplir las hipótesis de:

1. Distribución *uniforme*.
2. *Independencia* (no correlación serial).
3. Deben ser secuencias *largas* y sin huecos (*densas*).
4. Deben ser generadas por *algoritmos eficientes*.

## Método de Centros Cuadrados

Se elige un número, se lo eleva al cuadrado y luego se toman los dígitos del centro como el siguiente número; luego se repite el procedimiento.

Ejemplo:

2061 → 4247721

2477 → 6135529

1355 → . . .

Desventaja: la secuencia generada generalmente es corta.

En este ejemplo se generan 34 números pasando luego a sortear siempre 0. El 2500 genera siempre el 2500. En algunos casos, con grandes números se puede llegar a generar secuencias de 100.000 números diferentes.



## Método Congruencial Lineal (MCL)

Es un método ampliamente utilizado.

Se basa en la siguiente recurrencia:

$$n_i = (a \times n_{i-1} + c) \bmod m = f(n_{i-1})$$

Si se quiere obtener números uniformes en  $(0,1)$  se normaliza el resultado:

$$U_i = n_i/m$$

# Método Congruencial Lineal

Ejemplos:

- $a = 3, c = 0, m = 5, n_0 = 4$  genera la secuencia 2,1,2,4,2,1,...
- $a = 3, c = 0, m = 9, n_0 = 4$  genera la secuencia 3,0,0,...

En el MCL, si se repite un número, se repetirá toda la secuencia.

Ventajas:

- utiliza poca memoria y es muy rápido,
- fácil de volver a generar la misma secuencia, almacenando un solo número (alcanza con partir de la misma *semilla*  $n_0$ ).

# Método Congruencial Lineal

*Importante:* la velocidad de generación y por sobre todo el largo de la secuencia dependen de la elección de las constantes  $a$ ,  $c$ ,  $m$  y la semilla  $n_0$ .

Reglas que aseguran un *ciclo maximal* (Knuth<sup>1</sup>):

1.  $c$  y  $m$  deben ser primos relativos,
2. si  $p$  es factor primo de  $m$ , entonces elegir  $a = 1$ ,
3. si 4 es factor de  $m$ , elegir  $a = 1$ .

---

<sup>1</sup>Knuth, D., The Art of Computer Programming Vol. 2: Seminumerical Algorithms, Addison Wesley, 1998.

# Método Mersenne Twister (MT)

- Los generadores en base al Método Congruencial Lineal son muy utilizados, pero muchos de ellos tienen un período mucho más corto que el que uno desearía o necesita.
- En 1998, Matsumoto y Nishimura<sup>2</sup> presentaron los generadores MT, que son de período largo y rápidos.
- Existen varias implementaciones; es utilizado para Simulación a Eventos Discretos y Monte Carlo.

---

<sup>2</sup>Matsumoto, M.; Nishimura, T., Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Transactions on Modeling and Computer Simulation 8(1): 3-30,1998.

## Método Mersenne Twister

Genera números pseudoaleatorios utilizando la siguiente recurrencia:

$$X_{k+n} = X_{k+m} \oplus (X_k^u | X_{k+1}^l)A, \quad (k = 0, 1, \dots) \text{ donde:}$$

- $X$ : entero de  $w$  bits,
- $\oplus$ : XOR,
- $|$ : concatenación de cadenas de bits,
- $X^j$ : fragmento de  $j$  bits de  $X$ ;  $u$  son los bits más significativos y  $l$  los menos significativos,
- $A$ : matriz de  $w \times w$ ,
- $n$ : grado de recurrencia con  $1 \leq m \leq n$ .

# Método Mersenne Twister

- El método trabaja a más bajo nivel dado que utiliza operaciones lógicas en cadenas de bits, entonces es un generador rápido.
- Mt19937 es un ejemplo, tiene un período de  $2^{19937} - 1$ .
- El método ha sido validado y ha superado tests exigentes (por ejemplo, Dieharder<sup>3</sup>).

Más información en [http://en.wikipedia.org/wiki/Mersenne\\_twister](http://en.wikipedia.org/wiki/Mersenne_twister)

Página web <http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html>

---

<sup>3</sup><http://webhome.phy.duke.edu/~rgb/General/dieharder.php>

# Torrentes

- Un generador de números pseudoaleatorios que comience con la *misma semilla*, siempre producirá el mismo torrente o secuencia de números.
- *Diferentes semillas* generarán diferentes secuencias. Si las semillas se eligen con valores no cercanos (en el ciclo del generador), entonces las secuencias de números generados (torrentes) parecerán y actuarán como números aleatorios independientes entre sí, con lo que colaborarán en la generación de variables aleatorias independientes entre sí.

## Torrentes

- Al comparar los efectos de distintas políticas en un modelo (por ejemplo, número de camas), es importante que las corridas del modelo se ejecuten con los mismos valores (tiempos) en las actividades y las variables de decisión (dados por muestras de variables aleatorias).
- Cuando generamos muestras de una variable aleatoria, si utilizamos la misma secuencia de números pseudoaleatorios, generamos la misma secuencia de valores (muestras) de esas variables aleatorias.
- Para obtener valores esperados de las variables aleatorias, se realizan varias corridas de la simulación. En cada corrida deben ser usados diferentes torrentes de números e independientes entre sí ( $n_0$  distinto para cada variable aleatoria). El valor de la muestra (resultado) debe ser dado en un intervalo de confianza.



# Generación de números pseudoaleatorios en Java

Clase Random disponible en `java.util`.

`Random r = new Random(semilla);` crea un generador y lo inicializa con una semilla (valor entero dado).

`Random r = new Random();` crea un generador y lo inicializa con una semilla probablemente muy diferente a otra utilizando la misma función anteriormente.

`r.setSeed();` setea o actualiza la semilla de un generador ya creado y eventualmente ya utilizado.

`r.nextDouble();` retorna el siguiente número pseudoaleatorio del torrente en el intervalo real abierto (0,1).

# Test de Uniformidad e Independencia

Test de  $\chi^2$

Utilizado para analizar la uniformidad de la secuencia de los números pseudoaleatorios (válido para ajustar otras funciones de distribución).

El método consiste en tomar  $n$  observaciones independientes de la variable aleatoria (en nuestro caso los números generados), que llamaremos

$$X_1, X_2, \dots, X_n$$

## Test de $\chi^2$

El intervalo en el que varía la variable aleatoria se divide en  $k$  categorías.

Se conoce la probabilidad (teórica)  $P_s$  de que la variable aleatoria muestree en cada categoría  $s$ .

Sea  $Y_s$  la cantidad de valores de  $X_i$  pertenecientes a la categoría  $s$ .

Entonces:

$$Y_1 + Y_2 + \dots + Y_k = n$$

$$P_1 + P_2 + \dots + P_k = 1$$

## Test de $\chi^2$

El estimador

$$V = \sum_{s=1}^k \frac{(Y_s - nP_s)^2}{nP_s}$$

es una variable aleatoria con distribución  $\chi^2$  con  $k - 1$  grados de libertad ( $n$  debe ser grande para que el test sea válido).

Además se debe cumplir que  $nP_s > 5$  (al menos 5 observaciones en cada categoría).

## Test de $\chi^2$

Se calcula  $V$  y se analiza el mismo utilizando la tabla de  $\chi^2$  (tomar de libro, planilla electrónica o internet) mediante el test de significación siguiente:

- Si  $V > \chi_{k-1}^2$  (valor de la tabla), rechazar la hipótesis (de independencia).
- Si no, no rechazar la hipótesis.

Dado un nivel de significación (por ejemplo de  $\alpha = 0,05$  equivalente al 95%), nos fijamos en la tabla el valor correspondiente a ese nivel y para  $k - 1$  grados de libertad ( $\chi_{\alpha, k-1}^2$ ).

Si  $V > \chi_{\alpha, k-1}^2$  se rechaza la hipótesis y si no, no se rechaza.

## Ejemplo

Tirando 96 veces un dado se obtuvieron cantidades de 1s, 2s, etc: 15, 7, 9, 20, 26, 19. Se desea saber si el dado es simétrico, la hipótesis es que  $P_1 = P_2 = \dots P_6 = 1/6$ .

En este caso,  $n = 96$ ,  $Y_1 = 15$  etc.,  $nP_i = 16$ ,  $k - 1 = 5$ .

$$V = (15 - 16)^2/16 + (7 - 16)^2/16 + \dots (19 - 16)^2/16 = 16$$

Fijando un nivel de significación  $\alpha = 0.1$  tenemos  $V = 16 > 15.1 = \chi_{\alpha, k-1}^2$

Por lo tanto se rechaza la hipótesis (se supone que el dado no es simétrico).

## Test serial

Se utiliza para analizar la correlación serial de la secuencia de números observados.

Se agrupa la muestra en pares de valores los cuales estarán distribuidos uniforme e independientemente, en caso de verificar el test.

La muestra será de  $2n$  valores.

Se consideran las  $n$  parejas  $(X_{2j}, X_{2j+1})$  con  $0 < j < n$

$$(X_0, X_1)(X_2, X_3) \dots (X_{2n-2}, X_{2n-1})$$

## Test serial

Cada pareja tiene probabilidad  $P(X_{2j}, X_{2j+1}) = 1/k^2$  y tendremos  $k^2$  categorías (que es la cantidad de combinaciones posibles de parejas de números uniformes).

Se aplica el test de  $\chi^2$  con  $k^2 - 1$  grados de libertad con el mismo criterio que definimos anteriormente.



## Sorteos de variables aleatorias

- Variables aleatorias *discretas*: General, Poisson.
- Variables aleatorias *continuas*: Exponencial negativa, Normal, log Normal.

## Distribución discreta (general)

$X$  es un variable aleatoria que puede tomar los valores  $x_1, x_2, \dots, x_n$ , con probabilidad respectiva  $p_1, p_2, \dots, p_n$ .

Dividimos el intervalo  $[0,1]$  en intervalos de longitud  $p_1, p_2, \dots, p_n$ , donde se cumple  $\sum_{i=1}^n p_i = 1$ .

Al intervalo  $i$  le corresponden aquellos valores  $x_i$  que cumplen

$$\sum_{j=0}^{i-1} p_j \leq x_i \leq \sum_{j=0}^i p_j$$

## Distribución discreta (general)

Sorteo de una muestra de  $X$ :

1. Se toma un número  $u$  con distribución uniforme en  $(0,1)$ .
2. Si  $u$  cae en el intervalo correspondiente al número  $i$ , entonces aceptamos que  $X = x_i$  en este sorteo.

El método es válido, dado que la probabilidad de que  $u$  pertenezca a uno de los intervalos es igual a su longitud, ya que  $u$  es uniforme en  $(0,1)$ :

$$P(0 < u < p_1) = p_1$$

$$P(p_1 < u < p_1 + p_2) = p_2$$

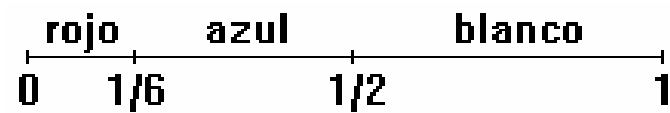
...

$$P(1 - p_n < u < 1) = p_n$$

## Ejemplo

Una variable aleatoria  $X$  puede tomar los valores: rojo con probabilidad  $1/6$ , azul con  $1/3$  y blanco con  $1/2$ .

Construimos el esquema



Si  $u = 0,1212 \rightarrow$  rojo

Si  $u = 0,9432 \rightarrow$  blanco

Si  $u = 0,6111 \rightarrow$  blanco

Si  $u = 0,4343 \rightarrow$  azul

## Distribución Poisson

Generalmente utilizada para modelar la cantidad de arribos en un intervalo de tiempo.

Función de probabilidad  $f(x) = \lambda^x \exp(-\lambda) / x!$  modela la probabilidad de  $x$  arribos en un intervalo de tiempo consistente con la tasa  $\lambda$  que expresa la cantidad media de arribos por unidad de tiempo.

Función de distribución cumple

- $F(x + 1) = F(x) + f(x + 1)$
- $F(0) = f(0) = \exp(-\lambda)$

propiedad que se utiliza para la obtención de muestras (ver pseudocódigo en Davies y O'Keefe).

## Generación de variables aleatorias continuas

Queremos generar valores de  $X$ , variable aleatoria con distribución

$$F_X(x) = \int_{-\infty}^x f_X(t) dt.$$

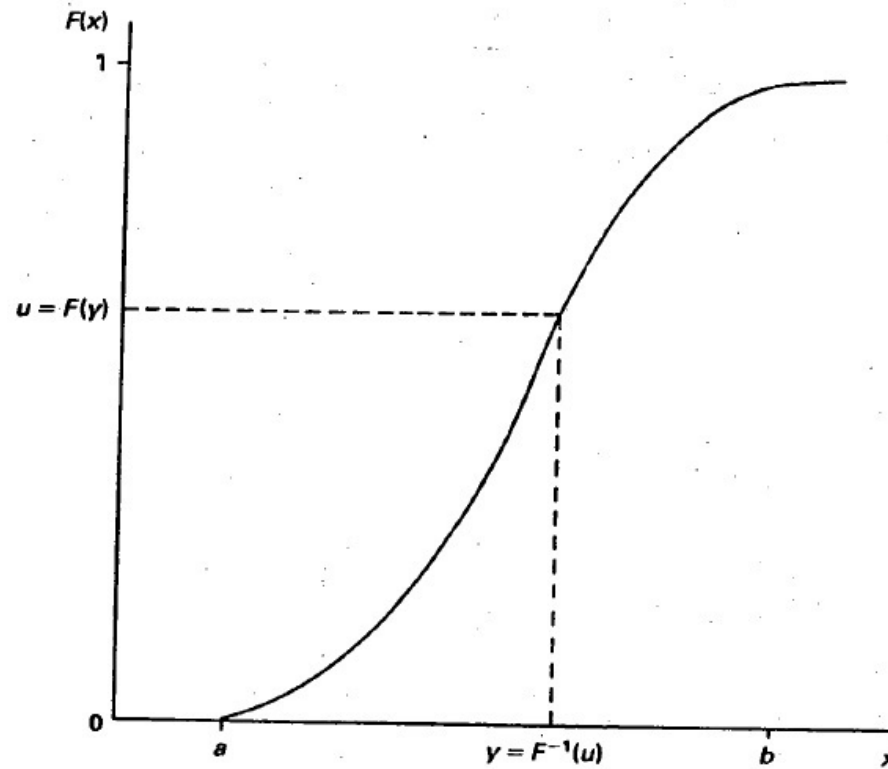
Método de la *transformación inversa*: se iguala la función de distribución de  $X$  a una variable aleatoria uniforme en  $(0,1)$ :

$$U = F_X(X) \Rightarrow X = F_X^{-1}(U)$$

El problema se reduce a encontrar una expresión analítica de la función inversa, de modo de despejar  $X$  en función de  $U$ .

## Generación de variables aleatorias continuas

Si  $F(x)$  es continua y estrictamente creciente,  $0 < F(x) < 1$ , entonces la función inversa siempre existe. Interpretación geométrica:



## Distribución Uniforme

Queremos obtener muestras de una variable aleatoria uniforme  $U(a, b)$ , por lo tanto  $f(x) = 1/(b - a)$ .

$$F_X(x) = \int_a^x \frac{1}{b-a} dt \text{ y además tenemos } u = \int_a^x \frac{1}{b-a} dt$$

Entonces  $u = (x - a)/(b - a)$ , por lo tanto  $x = a + u(b - a)$

Finalmente,  $x$  es  $U(a, b)$  siendo  $u$  una variable aleatoria  $U(0, 1)$  (número pseudoaleatorio).



## Distribución Exponencial Negativa

Generalmente usada para modelar el tiempo entre dos sucesos Poisson.

$$F_X(x) = 1 - \exp(-\lambda x) \Rightarrow u = 1 - \exp(-\lambda x) \Rightarrow \exp(-\lambda x) = 1 - u$$

$$-\lambda x = \ln(1 - u) \Rightarrow x = (-1/\lambda)\ln(u)$$

$(1 - u)$  es equivalente a  $u$ , ya que es  $U(0, 1)$ .

Entonces  $x = (-1/\lambda)\ln(u)$  es una muestra de una variable aleatoria exponencial de parámetro  $\lambda$ , a partir de una muestra de una variable aleatoria  $U(0, 1)$  (número pseudoaleatorio).

## Distribución Normal

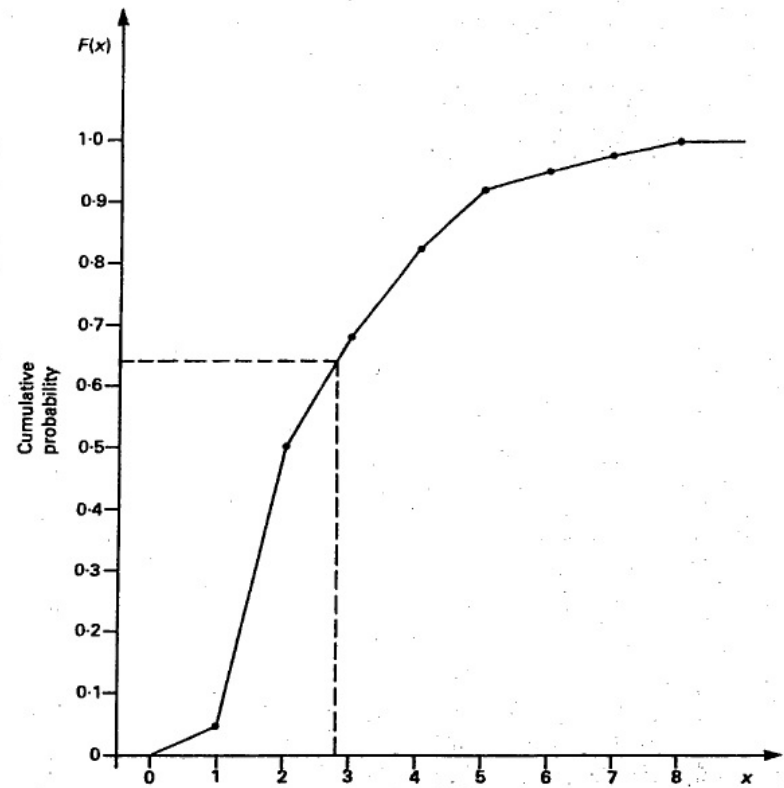
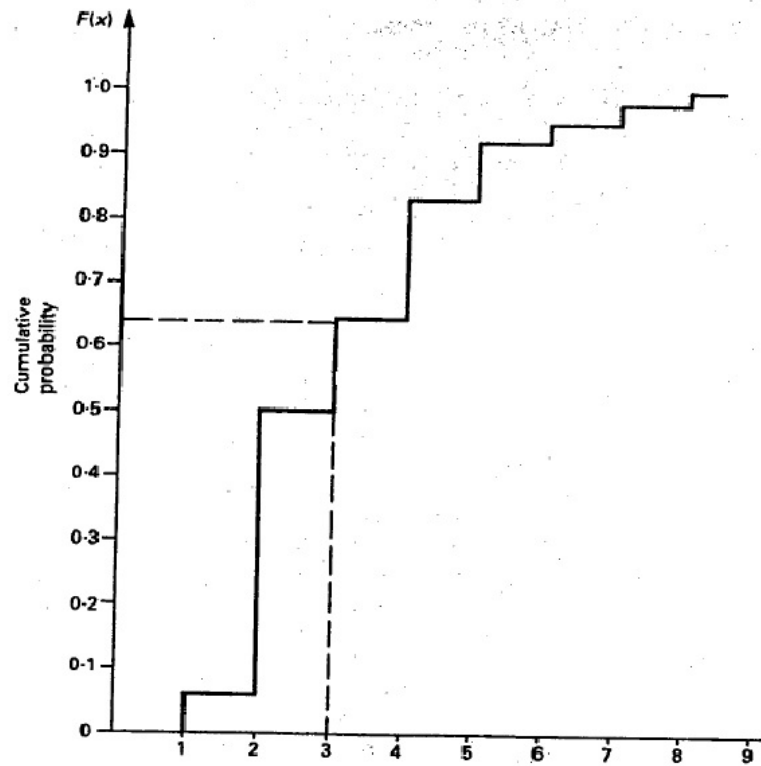
Usada para modelar la duración de actividades que varían estocásticamente alrededor de un valor medio, por ejemplo, el tiempo que lleva arreglar una máquina con una falla bien conocida. Tiene dos parámetros: media  $\mu$  y desviación estándar  $\sigma$ .

Observar que esta distribución no es la más adecuada para valores de tiempos, ya que al no tener cota inferior, puede tomar valores negativos (los tiempos son positivos y tienen valor mínimo cero). Si se utiliza, deben descartarse los valores negativos. Alternativa: utilizar la distribución logNormal.

Dado que no existe una expresión analítica cerrada para la inversa de la función de distribución de la Normal, para generar valores se utiliza el método de Box y Muller (ver Davies y O'Keefe) u otros de la literatura.

# Distribuciones empíricas arbitrarias

Variantes discreta y continua. Ver ejemplo clase anterior, última diapositiva.



## Distribuciones empíricas arbitrarias

Método de transformación inversa. Algoritmo en Davies y O'Keefe (hay una corrección):

$u$  = número pseudoaleatorio en el rango  $(0,1)$

$x = 1$

mientras ( $u$  es mayor que  $F(x)$ )

$x = x + 1$

Si es discreta devolver el valor

$x - 1$

Si es continua devolver el valor

$x - 1 + (u - F(x-1))/(F(x)-F(x-1))$