

UMA INTRODUÇÃO AOS MÉTODOS DE GERAÇÃO DE NÚMEROS E VARIÁVEIS ALEATÓRIAS PARA APLICAÇÕES EM SIMULADORES

A geração de números e variáveis aleatórias é um ingrediente fundamental em qualquer programa de simulação, comercial ou não. Na grande maioria dos programas comerciais, este processo é transparente para o usuário, restando a este apenas definir os parâmetros desta ou daquela distribuição de probabilidades desejada. Este texto introdutório está voltado para o leitor que pretende desenvolver programas de simulação com base em uma linguagem de propósito geral ou para aqueles desejosos de uma melhor compreensão dos conceitos e técnicas envolvidas nos processos de geração de números e variáveis aleatórias que se encontram presentes nas linguagens específicas de simulação.

Tópicos

- 1.1 Propriedades dos Números Aleatórios
- 1.2 Propriedades Desejadas aos Geradores de Números Aleatórios
- 1.3 Métodos de Geração de Números Aleatórios
- 1.4 Geração de Variáveis Aleatórias
- 1.5 Sumário
- Referências Bibliográficas

1.1 Propriedades dos Números Aleatórios

Uma sequência de números aleatórios, x_1, x_2, \dots , deve possuir duas importantes propriedades: uniformidade e independência. Todo número aleatório x_i é uma amostra independente de uma distribuição uniforme e contínua no intervalo de zero a 1. Desta forma, a função densidade de probabilidade de x é dada por:

$$f(x) = \begin{cases} 1, & 0 \leq x \leq 1 \\ 0, & \text{outro valor} \end{cases}$$

A função densidade de probabilidade é mostrada na Figura 1.

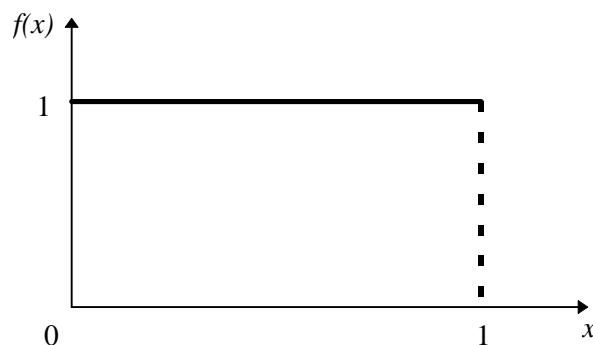


Figura 1: FDP para números aleatórios

O valor esperado para cada x_i é dado por:

$$E(x) = \int_0^1 x dx = \frac{x^2}{2} \Big|_0^1 = \frac{1}{2}$$

A variância por sua vez é dada por:

$$V(x) = \int_0^1 x^2 dx - [E(x)]^2 = \frac{x^3}{3} \Big|_0^1 - \left(\frac{1}{2}\right)^2 = \frac{1}{12}$$

Algumas das consequências da uniformidade e independência são as seguintes:

1. Se o intervalo de $[0, 1]$ é subdividido em n classes, ou subintervalos de igual tamanho, o valor esperado de observações em cada intervalo será de N/n , onde N é o número total de observações.
2. A probabilidade de observar um valor em um particular intervalo é independente dos valores previamente obtidos.

1.2 Propriedades Desejadas aos Geradores de Números Aleatórios

Para melhor compreender por que existem vários métodos geradores de números aleatórios e por que alguns são considerados melhores do que outros, exemplifica-se como estes geradores operam. A técnica empregada mais comum faz uso de uma relação recursiva na qual, o próximo número na sequência é uma função do último ou dois últimos números gerados, isto é,

$$x_n = f(x_{n-1}, x_{n-2}, \dots)$$

Um exemplo desta função é:

$$x_n = 5 x_{n-1} + 1 \text{ mod } 16$$

Iniciando a série com $x_0 = 5$, obtemos x_1 da forma que segue:

$$x_1 = 5(5) + 1 \text{ mod } 16 = 26 \text{ mod } 16 = 10$$

Os primeiros 32 números obtidos por meio deste procedimento são: 10, 3, 0, 1, 6, 15, 12, 13, 2, 11, 8, 9, 14, 7, 4, 5, 10, 3, 0, 1, 6, 15, 12, 13, 2, 11, 8, 9, 14, 7, 4, 5.

Observa-se que os valores de x são inteiros entre 0 e 15. Dividindo-os por 16, obtém-se uma sequência de números aleatórios com valores entre 0 e 1. Para o exemplo acima os números serão:

0,6250	0,1875	0,0000	0,0625	0,3750	0,9375	0,7500	0,8125
0,1250	0,6875	0,5000	0,5625	0,8750	0,4375	0,2500	0,3125
0,6250	0,1875	0,0000	0,0625	0,3750	0,9375	0,7500	0,8125
0,1250	0,6875	0,5000	0,5625	0,8750	0,4375	0,2500	0,3125

Fica claro que, conhecida a função f , pode-se gerar novamente a sequência sempre que se fornece o valor inicial de x_0 . Este valor, usado para iniciar a sequência, é conhecido por *semente*.

Uma importante observação sobre o exemplo dado, é que a função f é determinística. Desta forma, dada uma semente, pode-se afirmar, com 100% de certeza, qual serão os números na sequência. Embora estes números sejam considerados randômicos, no sentido de serem aprovados em testes estatísticos de aleatoriedade, são, de fato, *pseudo-aleatórios*. Embora isto seja verdadeiro, o objetivo em qualquer método de geração é produzir uma sequência de números aleatórios entre zero e 1, a qual possua propriedades semelhantes aquelas dos verdadeiros números aleatórios. Além disso, os números pseudo-aleatórios muitas vezes apresentam vantagens sobre os *verdadeiramente-aleatórios*. Por exemplo, quando se trata de empregá-los em simulações nas quais é desejável a possibilidade de se repetir o experimento simulado e, portanto, a sequência de números aleatórios, da maneira exata como foi executada anteriormente. É claro que se o interesse for uma sequência diferente, pode-se, a qualquer momento, fazer uso de valores diferentes para a semente. Desta forma, os geradores de números aleatórios nos fornecem um controle adicional sobre a possibilidade de reproduzir os resultados.

Outra importante característica evidenciada no exemplo apresentado, é que somente os 16 primeiros valores são únicos. O 17º é igual ao primeiro e o restante da sequência é apenas uma repetição cíclica dos primeiros 16 números. Dito de outra forma, o gerador utilizado possui um *comprimento de ciclo* igual a 16 valores. Alguns geradores não repetem uma parte inicial do ciclo, chamada de *cauda*. Neste caso, o comprimento de seu *período* é dado pela soma do comprimento L da cauda mais o comprimento C do ciclo (Figura 2).

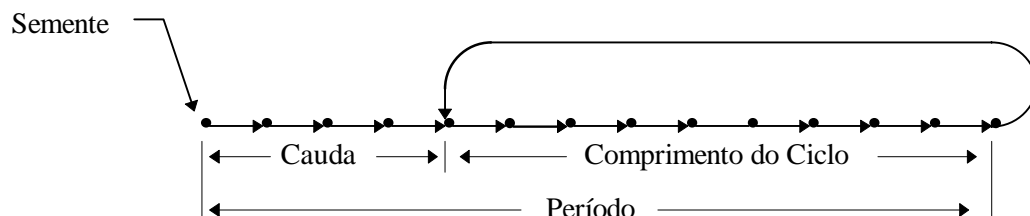


Figura 2: Comprimento do ciclo, cauda e período de gerador de números aleatórios

As propriedades desejadas em um gerador de números aleatórios são as seguintes:

1. *Deve ser computacionalmente eficiente:* Uma vez que as simulações podem necessitar da geração de, até mesmo, milhões de números aleatórios em cada execução, o tempo para processar cada geração deve ser mínimo;
2. *O período deve ser muito longo:* Um período curto pode fazer com que haja a reciclagem da sequência de números aleatórios, resultando em uma repetição da sequência de eventos. Como consequência, pode haver uma limitação do período útil de uma rodada de simulação.
3. *Os sucessivos valores devem ser independentes e uniformemente distribuídos:* A correlação entre os diversos valores gerados deve ser pequena. A correlação, se significativa, indica dependência.

A duas primeiras propriedades são facilmente alcançáveis. A terceira, no entanto, requer uma série de testes estatísticos para garanti-la. A literatura sob o tema [9] oferece e analisa uma

série de métodos geradores de números aleatórios consagrados. Neste texto introdutório, se trata, principalmente, do mais utilizado deles, isto é, o método Congruente Linear.

1.3 Métodos de Geração de Números Aleatórios

O método Congruente Linear é considerado o mais popular, entre tantos outros métodos geradores de números aleatórios e, por esta razão, será tratado com um pouco mais de detalhes. Além deste, se tratará também de algumas de suas extensões, às quais são reportados seqüências com longos períodos. Muitos outros métodos poderão ser encontrados nas referências como, por exemplo, em Law e Kelton (1991) e Bratley, Fox e Schrage (1987).

1.3.1 Método Congruente Linear (MCL)

Este método, também conhecido como *método congruente misto*, foi primeiramente divulgado em um trabalho desenvolvido pelo Prof. D. H. Lehmer, em 1951, quando dos experimentos executados pelo computador ENIAC no MIT, conforme citado por Jain, [1991]. Em suas pesquisas ele descobriu que restos de sucessivas potências de um número possuem boas características de aleatoriedade. Ele obtinha o *n-ésimo* número de uma seqüência, tomando o resto da divisão da *n-ésima* potência de um inteiro *a* por um outro inteiro *m*. Isto é:

$$x_n = a^n \bmod m$$

Uma expressão equivalente usada para o cálculo de x_n após calcular x_{n+1} é dada por:

$$x_n = ax_{n-1} \bmod m$$

Os parâmetros *a* e *m* são chamados de *multiplicador* e *módulo* respectivamente. Ainda segundo Jain (1991), as escolhas de Lehmer para estes parâmetros foram $a = 23$ e $m = 10^8 + 1$. Tais valores foram baseados na facilidade de implementação no ENIAC, que era uma máquina de oito dígitos decimais.

Muitas das propostas atuais são generalizações da proposta de Lehmer e seguem a seguinte fórmula:

$$x_n = ax_{n-1} + b \bmod m$$

Os valores de x_n são inteiros entre 0 e $m-1$. As constantes *a* e *b* são positivas.

A popularidade dos geradores baseados neste método deve-se ao fato de serem facilmente analisados e de algumas garantias de suas propriedades dadas pela teoria das congruências [Dudewicz e Karian (1985)].

De maneira geral, a escolha dos valores de *a*, *b*, e *m* afeta o período e a autocorrelação na seqüência. Vários pesquisadores estudaram tais influências. O resumo destes resultados são apresentados nas referências Jain (1991), Banks (1996), Law (1991) e [Dudewicz e Karian (1985)]., e são apresentados abaixo:

1. O módulo de *m* deve ser grande. Uma vez que os valores de *x* estarão entre 0 e $m-1$, o período nunca será maior do que *m*;

2. Para que a computação de $\text{mod } m$ seja eficiente, m deve ser uma potência de 2, isto é, 2^k . Neste caso, o $\text{mod } m$ poderá ser obtido truncando-se o resultado à direita por k bits.
3. Se b for diferente de zero, o máximo período possível m é obtido se e somente se:
 - a) os inteiros m e b sejam primos, um em relação ao outro, isto é, não possuam nenhum outro fator além de 1;
 - b) todo número primo que é um fator de m , é também um fator de $a-1$;
 - c) $a-1$ é um múltiplo de 4, se o inteiro m é múltiplo de 4.
4. Se $b = 0$, e m potência de 2, o maior período possível será $P = m / 4$, considerando que: x_0 (semente) seja um número ímpar e o multiplicador (a) seja dado por $a = 8k + 3$ ou $a = 8k + 5$, para algum $k = 0, 1, 2, \dots$

Observe que todas estas condições são alcançadas se $m = 2^k$, $a = 4c + 1$ e $b = \text{ímpar}$. Neste caso, c , b e k são inteiros positivos.

Um gerador que possua o maior período possível é chamado de *gerador de período completo*. Nem todos os geradores de período completo são igualmente bons. A questão da autocorrelação deve também ser considerada. Aqueles com baixa correlação são, obviamente, preferidos. O exemplo abaixo revela tais diferenças. Os dois geradores possuem o período completo, mas o primeiro apresenta uma correlação de 0,25 entre x_{n-1} e x_n , enquanto que no segundo esta correlação é menor do que 2^{-18} .

$$X_n = (2^{34} + 1)x_{n-1} + 1 \text{ mod } 2^{35}$$

$$X_n = (2^{18} + 1)x_{n-1} + 1 \text{ mod } 2^{35}$$

Vejamos um exemplo de geração de números aleatórios usando o MCL.

Exemplo 1

Use o MCL para gerar uma sequência de números aleatórios entre zero e 1, com os seguintes parâmetros: $x_0 = 27$, $a = 17$, $b = 43$ e $m = 100$: Observe que os valores inteiros gerados, serão todos entre zero e 99, em razão do módulo. Observe também, que estarão sendo gerados inteiros aleatórios e não números aleatórios. Tais inteiros podem ser transformados em números aleatórios (R_i 's) entre zero e 1, aplicando-se a relação:

$$R_i = x_i / m, i = 1, 2, \dots$$

A sequência de valores para x_i e subsequentes R_i , é apresentada abaixo:

$$x_0 = 27$$

$$x_1 = (17 \cdot 27 + 43) \text{ mod } 100 = 502 \text{ mod } 100 = 2$$

$$R_1 = 2 / 100 = 0,02$$

$$x_2 = (17 \cdot 2 + 43) \text{ mod } 100 = 77 \text{ mod } 100 = 77$$

$$R_2 = 77 / 100 = 0,77$$

$$x_3 = (17 \cdot 77 + 43) \text{ mod } 100 = 1352 \text{ mod } 100 = 52$$

$$R_3 = 52 / 100 = 0,52$$

.

1.3.2 Método Congruente Linear Multiplicativo (MCLM)

Uma das derivações do MCL é o método congruente linear multiplicativo. Neste método, o valor do incremento $b = 0$. Desta forma, o gerador fica reduzido a seguinte expressão:

$$x_n = ax_{n-1} \bmod m$$

Computacionalmente falando, os geradores baseados no MCLM são mais eficientes do que aqueles com base no MCL. Uma vez que não existe o envolvimento de adições, o tempo de processamento necessário se reduz. Tal eficiência pode ser ainda maior quando m assume uma potência de 2, fazendo com que, desta forma, a operação mod seja trivial. Podem-se destacar dois métodos multiplicativos derivados do MCLM, aqueles com $m = 2^k$ e aqueles com $m \neq 2^k$. Abaixo um exemplo empregando o MCLM com $m = 2^k$.

Exemplo 2

Encontre o período para o gerador com os seguintes parâmetros: $a = 13$, $m = 2^6$, e $x_0 = 1, 2, 3$ e 4. A solução é dada na tabela 1.

i	x_i	x_i	x_i	x_i
0	1	2	3	4
1	13	26	39	52
2	41	18	59	36
3	21	42	63	20
4	17	34	51	4
5	29	58	23	
6	57	50	43	
7	37	10	47	
8	33	2	35	
9	45		7	
10	9		27	
11	53		31	
12	49		19	
13	61		55	
14	25		11	
15	5		15	
16	1		3	

Tabela 1: Variação dos períodos para várias sementes

Observa-se que com as sementes ímpares (1 e 3), é possível a obtenção de períodos com 16 elementos ($P = m/4 = 64/4 = 16$). Para as sementes pares (2 e 4), os períodos obtidos têm comprimentos 8 e 4, respectivamente. Observa-se, também que a obedece a fórmula $8k + 5$, com $k = 1$, exigida para o alcance de períodos máximos.

Afirmou-se anteriormente, que qualquer método gerador de números aleatórios, incluindo o MCL, deve gerar uma sequência de valores uniformes e independentes. Além de tais

propriedades, necessárias a um bom gerador, algumas propriedades secundárias também são requeridas. Dentre estas se salienta a da máxima densidade, isto é, a necessidade de que os valores assumidos por R_i , $i = 1, 2, \dots$, não deixem grandes folgas ou “buracos” no intervalo $[0, 1]$. O espaço entre os valores de R_i deve ser uniforme e mínimo. No caso do gerador deste exercício, tais folgas são demasiadamente grandes. Os valores gerados para $x_0 = 1$ são $\{1, 5, 9, 13, \dots, 53, 57, 61\}$. A folga neste caso pode ser calculada pela relação entre dois valores consecutivos, isto é: $folga = 5/64 - 1/64 = 0,0625$. Este é um valor considerado muito grande para a maioria das aplicações. O gerador exemplificado não é viável para, praticamente, qualquer tipo de aplicação devido ao curto período e a grande folga. O exemplo, no entanto, demonstra os cuidados necessários para com os parâmetros dos geradores.

A propósito, velocidade e eficiência computacional são elementos sempre da maior importância quando se trata do uso de computadores digitais. O exemplo a seguir, adaptado de Banks (1996), mostra que uma boa escolha do *módulo* sempre traz benefícios neste sentido.

Exemplo 3

Velocidade e eficiência sempre são beneficiadas quando a escolha de m é uma potência de 2 ou muito próximo disso. Uma vez que a maioria dos computadores digitais usa uma representação binária dos números, a operação de cálculo do *resto* é sempre conduzida mais eficientemente quando $m = 2^k$. Após o cálculo de $ax_i + b$, x_{i+1} é obtido pela retirada do dígito binário mais a esquerda de $ax_i + b$ e pelo uso do k -ésimo dígito binário mais a direita.

Um exemplo proposto por Banks (1996), usando o formato decimal, (de mais fácil compreensão), ilustra, por analogia, tal mecanismo. Assume-se que $m = 10^2 = 100$, $a = 19$, $c = 0$ e $x_0 = 63$ e gera-se uma série de números aleatórios usando a equação do MLC.

$$x_0 = 63$$

$$x_1 = (19)(63) \bmod 100 = 1197 \bmod 100 = 97 \rightarrow (1197/100 = 11,97. \text{ k=2 dígitos à direita} = 97)$$

$$x_2 = (19)(97) \bmod 100 = 1843 \bmod 100 = 43$$

$$x_3 = (19)(43) \bmod 100 = 817 \bmod 100 = 17$$

.

No caso m é uma potência de 10, isto é, $m = 10^k$ e a operação do módulo (ou de determinação do *resto*) é realizada tomando-se os k dígitos (decimais neste caso), mais à direita.

Para encerrar esta seção, apresenta-se um exemplo de um gerador ainda em uso em muitas rotinas para a geração de números aleatórios, presentes em programas comerciais. Os valores dos parâmetros satisfazem as condições para permitir $P = m - 1$ (maior do que 2 bilhões). Os valores dos parâmetros são: $m = 2^{31} - 1 = 2.147.483.647$ (que é um número primo), $a = 7^5 = 16.807$ e $c = 0$. A semente $x_0 = 123.456$. Os primeiros números gerados serão:

$$x_1 = (7^5)(123.456) \bmod (2^{31} - 1) = 2.074.941.799 \bmod (2^{31} - 1) = 2.074.941.799$$

$$R_1 = 2.074.941.799 / 2^{31} = 0,9662$$

$$x_2 = (7^5)(2.074.941.799) \bmod (2^{31} - 1) = 559.872.160$$

$$R_2 = 559.872.160 / 2^{31} = 0,2607$$

$$x_3 = (7^5)(559.872.160) \bmod (2^{31} - 1) = 1.645.535.613$$

$$R_3 = 1.645.535.613 / 2^{31} = 0,7662$$

•
•

Observe-se que nesta rotina a divisão é feita por $m + 1$, no lugar de m . No entanto, para grandes valores de m o efeito é insignificante.

Duas precauções importantes devem ser consideradas quando da implementação das rotinas computacionais referentes aos geradores aqui mencionados. A primeira delas diz respeito às propriedades dos métodos. Todo o processo computacional envolvido deve garantir tais propriedades. Para tanto, toda a computação deve ser feita de forma exata, sem arredondamentos. Em outras palavras, isto significa trabalhar com aritmética inteira sem erros de excesso ou “estouro” de capacidade (overflow). Em algumas linguagens a computação é realizada com números reais. Neste caso, os cuidados devem recair sobre os possíveis arredondamentos, os quais podem causar consideráveis reduções no período.

A segunda preocupação na implementação do MLC e suas derivações dizem respeito à grandeza do produto ax_{n-1} , o qual pode exceder a capacidade para o maior inteiro permitido no sistema. Uma possível alternativa para contornar este problema foi apresentada por Schrage (1979), citado por Jain (1991). A base do MLC foi apresentada na forma da seguinte identidade:

$$ax \bmod m = g(x) + mh(x)$$

onde

$$g(x) = (ax \bmod q) - r(x \operatorname{div} q)$$

e

$$h(x) = (x \operatorname{div} q) - (ax \operatorname{div} m)$$

onde, $q = m \operatorname{div} a$ e $r = m \bmod a$. A operação $A \operatorname{div} B$ é equivalente a dividir A por B truncando o resultado (isto é, removendo a parte fracionária do número). Pode ser mostrado que para todo os valores de x no intervalo $1, 2, \dots, m-1$, as expressões envolvendo $g(x)$ são todas menores do que $m-1$. Também pode ser mostrado que se $r < q$, $h(x)$ será 0 ou 1 e poderá ser inferido a partir de $g(x)$ e, $h(x)$ é 1 se e somente se $g(x)$ for negativo. Desta forma, a operação ax , que pode causar *overflow*, não precisará ser executada. O exemplo a seguir ilustra a destes conceitos.

Exemplo 4

Considere a implementação do seguinte MCLM:

$$x_n = 7^5 x_{n-1} \bmod (2^{31} - 1)$$

ou

$$x_n = 16.807 x_{n-1} \bmod 2.147.483.647$$

O produto de ax_{n-1} pode ser tão grande quanto $16.807 \times 2.147.483.647 \approx 1.03 \times 2^{45}$. A implementação deste gerador usando aritmética inteira produzira um *overflow*, a menos que o processador suporte inteiros de 46 bits ou mais. Neste caso, a implementação do método de Schrage, é feita da seguinte maneira:

$$a = 16.807$$

$$m = 2.147.483.647$$

$$q = m \operatorname{div} a = 2.147.483.647 \operatorname{div} 16.807 = 12.7773$$

$$r = m \operatorname{mod} a = 2.147.483.647 \operatorname{mod} 16.807 = 2.836$$

Jain (1991) apresenta uma rotina em PASCAL para este método, a qual é reproduzida abaixo:

```

FUNCTION Random (VAR x: INTEGER) : REAL;

CONST
    a = 16807;           (* Multiplicador *)
    m = 2147483647;      (* Módulo *)
    q = 127773;          (* m div a *)
    r = 2836;            (* m mod a *)

VAR
    x_div_q, x_mod_q, x_new: INTEGER;

BEGIN
    x_div_q := x DIV q;
    x_mod_q := x MOD q;
    x_new := a * x_mod_q - r * x_div_q;
    IF x_new > 0 THEN x := x_new ELSE x := x_new + m;
    Random := x/m;
END;

```

Segundo Jain (1991), esta rotina pode ser implementada em computadores cujo maior inteiro suportado seja $2^{31} - 1$ ou maior do que isso. O mesmo autor apresenta ainda uma rotina para números reais, caso o maior inteiro suportado pelo sistema seja menor do que o especificado. Um bom teste para uma rotina que implemente este método, é calcular o valor de $x_{10.000}$ iniciando com $x_0 = 1$. Uma correta implementação chegará ao valor 1.043.618.065.

1.4 Geração de Variáveis Aleatórias

Neste tópico trata-se dos métodos e procedimentos computacionais dedicados à geração de variáveis aleatórias com características específicas de alguma das diversas distribuições teóricas de probabilidades. A necessidade de tais variáveis pode ser atestada nos inúmeros exemplos de sistemas de filas e outros. Por exemplo, tempos entre chegadas, tempos de serviço ou demandas por produtos, são elementos muitas vezes de natureza aleatória e que necessitam serem incorporados aos modelos de simulação, mantendo estas características (Freitas, 2008). Uma série de procedimentos envolvendo técnicas de amostragem, estimação de parâmetros e testes de aderência, são necessários para que se possa bem determinar o tipo de distribuição teórica que determina o comportamento da(s) variável(eis) sob tratamento. Aqui, considera-se apenas a necessidade de geração computacional de tais variáveis. Todos os métodos baseiam-se na prévia geração de um número aleatório R , uniformemente distribuído sobre o intervalo $(0, 1)$. Para efeito de notação, x será uma variável aleatória com função densidade de probabilidade (fdp) $f(x)$, caso contínuo e $p(x)$ no caso discreto. Todos os métodos pretendem tratar o problema de expressar x como uma função explícita de R .

A literatura sobre o assunto reporta a existência de pelo menos cinco métodos básicos voltados para a geração de amostras aleatórias:

1. Transformação Inversa;
2. Transformação Direta;
3. Convolução;
4. Aceitação/Rejeição;
5. Propriedades Especiais;

O método a ser empregado na geração das variáveis aleatórias depende do tipo de distribuição e da eficiência que se está buscando no processo. Sobre as distribuições que serão tratadas a seguir, ora se fará uso de um ora de outro método, conforme os mesmos aparecem na literatura, reportando os dois critérios acima expostos. Neste texto introdutório ao assunto, se prefere não detalhar os métodos, em particular, mas mostrar a forma como se apresentam, quando empregados na obtenção de variáveis aleatórias que seguem esta ou aquela particular distribuição de probabilidades.

1.4.1 Geração de Distribuições Discretas

Dentre as muitas distribuições teóricas discretas possíveis de serem artificialmente geradas apresentam-se, particularmente, os procedimentos associados às distribuições: Geométrica, Poisson e Empírica Discreta.

1.4.1.1 Distribuição de Poisson

A distribuição de Poisson se caracteriza pela seguinte função densidade de probabilidade:

$$p(x) = P(X = x) = e^{-\lambda} \frac{\lambda^x}{x!} \quad x = 0, 1, 2, \dots, \lambda > 0$$

a qual representa a probabilidade de ocorrência de x sucessos, num dado intervalo de tempo. Onde λ , é o valor esperado do número de ocorrências por unidade de tempo.

Os procedimentos computacionais para a geração de uma variável aleatória Poisson, considerando a aplicação do método da Aceitação/Rejeição, são os seguintes:

1. Fazer $n = 0$ e $P = 1$;
2. Gerar um número aleatório R_{n+1} e substituir P por $P \cdot R_{n+1}$;
3. Se, $P < e^{-\lambda}$, aceitar $X = n$, caso contrário, rejeitar n atual, fazer $n = n + 1$, e retornar aos procedimentos no passo 2.

A idéia básica por trás do método da Aceitação/Rejeição, é gerar um número aleatório e testar uma determinada condição de “aceitação”. Caso esta condição seja satisfeita, o valor gerado deve ser aceito. Caso contrário, os passos devem ser repetidos. Verifica-se, claramente, a necessidade de se gerar mais números aleatórios do que o efetivamente utilizado, uma vez que, nem sempre os valores das variáveis aleatórias geradas, satisfarão à condição necessária. Abaixo um exemplo.

Exemplo 5

Gerar dois números, segundo uma distribuição de Poisson, com $\lambda = 0,2$. Primeiramente, computa-se o valor de $e^{-\lambda} = e^{-0,2} = 0,8187$. Na seqüência, obtém-se um conjunto de números aleatórios e se iniciam os procedimentos estabelecidos nos passos de 1 a 3 anteriormente firmados, os quais são apresentados na tabela 2.

	Passo	Geração de P , n e X
Geração do primeiro valor	1	$n = 0, P = 1$
	2	$R_1 = 0,4357; P = 1.R_1 = 0,4357$
	3	como $P = 0,4357 < e^{-0,2} < 0,8187$; aceita-se $X = 0$
	1 - 3	$R_1 = 0,4146$ leva a: $X = 0$
Geração do segundo valor	1	$n = 0, P = 1$
	2	$R_1 = 0,8353; P = 1.R_1 = 0,8353$
	3	como $P \geq e^{-\lambda}$; rejeita-se $n = 0$ e retorna-se ao passo 2 com $n = 1$
	2	$R_2 = 0,9952; P = P.R_2 = 0,8353.0,9952 = 0,8313$
	3	como $P \geq e^{-\lambda}$; rejeita-se $n = 1$ e retorna-se ao passo 2 com $n = 2$
	2	$R_3 = 0,8004; P = P.R_3 = 0,8313. 0,8004 = 0,6654$
	3	como $P = 0,6654 < e^{-0,2} < 0,8187$; aceita-se $X = 2$

Tabela 2: Valores para a variável aleatória Poisson

Como pode ser observado nos resultados, foram necessários 4 números aleatórios para gerar 2 variáveis de Poisson ($X = 0$ e $X = 2$). Numa longa simulação, com cerca de 1000 valores Poisson com $\lambda = 0,2$, serão necessários, aproximadamente $1.000(\lambda + 1)$ ou 1200 números aleatórios. Segundo Schrage (1987) e Law (1991), para valores maiores do que $\lambda = 15$, a ineficiência computacional é muito grande e uma aproximação pela distribuição normal torna-se computacionalmente mais adequada.

1.4.1.2 Distribuição Empírica Discreta

Uma distribuição empírica pode ser tanto discreta quanto contínua. Sua aplicação costuma estar associada à impossibilidade de determinação da distribuição teórica de probabilidades da variável aleatória sob estudo. Neste caso, ela é usada como uma aproximação da verdadeira distribuição.

Para gerar uma variável aleatória que tenha um comportamento semelhante ao determinado por distribuição empírica discreta conhecida, é necessário que, inicialmente, se determine as frequências relativas acumuladas da distribuição. Para exemplificar os procedimentos, imagine que uma amostragem realizada sobre determinada variável aleatória discreta tenha resultado nos valores da tabela 3:

x	$p(x)$	$F(x)$
0	0,50	0,50
1	0,30	0,80
2	0,20	1,00

Tabela 3: Valores da variável aleatória x e de suas probabilidades

A função densidade de probabilidade (fdp), $p(x)$, da variável é dada por:

$$\begin{aligned} p(0) &= P(X = 0) = 0,50 \\ p(1) &= P(X = 1) = 0,30 \\ p(2) &= P(X = 2) = 0,20 \end{aligned}$$

e a função densidade de probabilidade acumulada, $F(x) = P(X \leq x)$, é dada por

$$F(x) = \begin{cases} 0 & x < 0 \\ 0,5 & 0 \leq x < 1 \\ 0,8 & 1 \leq x < 2 \\ 1,0 & 2 \leq x \end{cases}$$

Uma vez que tais informações estejam disponíveis, aplica-se o método da transformação inversa que, neste caso, torna-se um processo de pesquisa em uma tabela de valores, num procedimento muito semelhante ao do método de Monte Carlo.

Exemplifica-se esta aplicação considerando a variável aleatória x da tabela 1.4. Suponha que se tenha gerado um número aleatório $R_1 = 0,73$. Traçando um gráfico da função densidade de probabilidade acumulada, verifica-se que para $F(x) = R_1$, o correspondente valor de x será 1, isto é, $X = 1$ (ver Gráfico 1). Logo, graficamente verifica-se que $R_1 = 0,73$ é transformado em $X = 1$.

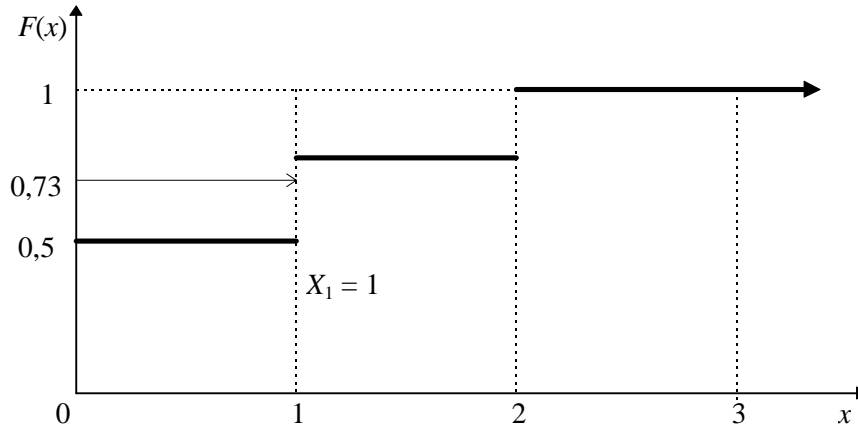


Gráfico 1: Transformação de $R_1 = 0,73$ em $X = 1$

Os procedimentos de busca são facilitados pela construção de uma tabela como a Tabela 4, abaixo.

i	Entrada r_i	Saída x_i
1	0,50	0
2	0,80	1
3	1,00	2

Tabela 4: Tabela de geração dos valores de X

Primeiramente, o objetivo é descobrir o intervalo no qual se encontra R_1 . Em geral, fazendo $R = R_1$, se,

$$F(x_{i-1}) = r_{i-1} < R \leq r_i = F(x_i)$$

então $X_1 = x_i$. Como $r_1 = 0,5 < R_1 = 0,73 \leq r_2 = 0,8$ então $X_1 = x_2 = 1$. O esquema de geração é resumido como segue:

$$X = \begin{cases} 0, & R \leq 0,5 \\ 1, & 0,5 < R \leq 0,8 \\ 2, & 0,8 < R \leq 1,0 \end{cases}$$

1.4.2 Geração de Distribuições Contínuas

Dentre as muitas distribuições teóricas contínuas, possíveis de serem artificialmente geradas, apresentam-se, particularmente, os procedimentos afetos as distribuições uniforme, triangular, exponencial e normal.

1.4.2.1 Distribuição Uniforme

Uma variável aleatória x tem distribuição uniforme sobre um intervalo $[a, b]$, se sua função densidade de probabilidade (fdp) é dada por:

$$f(x) = \frac{1}{b-a} \quad a \leq x \leq b$$

A técnica mais utilizada para a obtenção de uma variável aleatória uniformemente distribuída é a da transformação inversa. A fórmula é a seguinte:

$$x = a + (b - a)R$$

Os parâmetros necessários para a obtenção de uma variável com distribuição uniforme são apenas os valores extremos do intervalo $[a, b]$. Uma vez definidos, os seguintes passos devem ser considerados:

1. Gerar R ;
2. Calcular $x = a + (b - a)R$.

Exemplo 6

Gerar três valores de uma distribuição uniforme no intervalo $[10, 50]$. Usando uma tabela de valores aleatórios, obtém-se $R_1 = 0,932$; $R_2 = 0,105$ e $R_3 = 0,687$. Aplicando-se o método proposto obtém-se os resultados mostrados na Tabela 5:

Passo	Valor de R_i e de x_i
1	$R_1 = 0,932$
2	$x_1 = 10 + (40)0,932 = 47,28$
1	$R_2 = 0,105$
2	$x_2 = 10 + (40)0,105 = 14,2$
1	$R_3 = 0,687$
2	$x_3 = 10 + (40)0,687 = 37,48$

Tabela 5: Valores para a variável aleatória uniforme

1.4.2.2 Distribuição Triangular

Uma variável aleatória x tem uma distribuição triangular se sua fdp é dada por:

$$f(x) = \begin{cases} \frac{2(x-a)}{(b-a)(c-a)}, & a \leq x \leq b \\ \frac{2(c-x)}{(c-b)(c-a)}, & b < x \leq c \end{cases}$$

onde: $a \leq b \leq c$. A moda $b = 3 E(x) - (a + c)$.

Pelo método da transformação inversa facilmente se obtém a fórmula para gerar amostras com distribuição triangular. A variável x com esta distribuição é obtida por:

$$x = \begin{cases} a + \sqrt{R(b-a)(c-a)}, & \text{se } 0 \leq R \leq \frac{b-a}{c-a} \\ c - \sqrt{(1-R)(c-b)(c-a)}, & \text{se } \frac{b-a}{c-a} < R \leq 1 \end{cases}$$

Exemplo 7

Gerar três valores de uma distribuição triangular com parâmetros (0, 1, 2). Usando uma tabela de valores aleatórios, obtém-se $R_1 = 0,544$; $R_2 = 0,747$ e $R_3 = 0,449$. Aplicando o método proposto chega-se a:

$$x = \begin{cases} \sqrt{2R} & 0 \leq R \leq \frac{1}{2} \\ 2 - \sqrt{2(1-R)} & \frac{1}{2} < R \leq 1 \end{cases}$$

Os três valores obtidos são apresentados na Tabela 6:

Passo	Valor de R_i e de x_i
1	$R_1 = 0,544$
2	$x_1 = 2 - \sqrt{2(1-0,544)} = 1,045$
1	$R_2 = 0,747$

2	$x_2 = 2 - \sqrt{2(1 - 0,747)} = 1,288$
1	$R_3 = 0,449$
2	$x_3 = \sqrt{2(0,449)} = 0,947$

Tabela 6: Valores para a variável aleatória uniforme

1.4.2.3 Distribuição Exponencial

Uma variável aleatória x tem uma distribuição exponencial se sua fdp é dada por:

$$f(x) = \lambda e^{-\lambda x}, \quad x \geq 0$$

O parâmetro λ é interpretado como sendo o número médio de ocorrências por unidade de tempo, enquanto a razão $1/\lambda$ representa o tempo médio entre as ocorrências.

Aplicando-se o método da transformação inversa para a obtenção de uma variável aleatória x com distribuição exponencial resulta na seguinte relação:

$$x_i = -\lambda \ln(1 - R_i)$$

Uma vez que $(1 - R_i)$, da mesma forma que R_i , possui distribuição uniforme no intervalo $[0, 1]$, pode-se substituir $(1 - R_i)$ por R_i na expressão acima.

Exemplo 8

Gerar valores de uma distribuição exponencial com parâmetro $\lambda = 1$. De acordo com o método exposto acima, cada R_i gera um correspondente x_i conforme mostrado na Tabela 7.

i	1	2	3	4	5
R_i	0,1306	0,0422	0,6597	0,7965	0,7696
x_i	0,1399	0,0431	1,0779	1,5920	1,4679

Tabela 7: Valores para a variável aleatória exponencial

1.4.2.4 Distribuição Normal

Uma variável aleatória x tem uma distribuição normal se sua fdp é dada por:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad -\infty < x < \infty$$

Onde μ é a média da distribuição e σ é o desvio padrão.

O método utilizado para a obtenção de uma variável normal padronizada, isto é, com $\mu = 0$ e $\sigma = 1$, é conhecido como *método de Box-Muller*. Embora existam outras técnicas de obtenção

de variáveis normalmente distribuídas, este método é facilmente programável e apresenta bons resultados. Considere duas variáveis normais padronizadas, Z_1 e Z_2 , as quais correspondem às coordenadas de um ponto no plano (Gráfico 2).

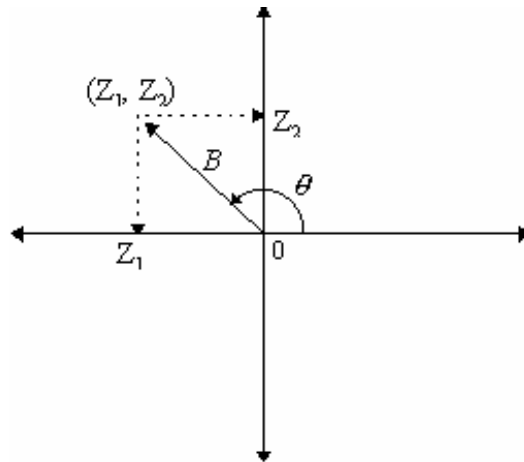


Gráfico 2: Representação polar do par de variáveis normais padronizadas

Este ponto pode ser representado em coordenadas polares:

$$\begin{aligned} Z_1 &= B \cos \theta \\ Z_2 &= B \sin \theta \end{aligned}$$

Após algumas transformações matemáticas, é possível demonstrar que o raio B pode ser obtido por:

$$B = (-2 \ln R)^{\frac{1}{2}}$$

Demonstra-se, também que o ângulo θ é uniformemente distribuído entre 0 e 2π radianos. A combinação destas relações nos fornece as duas equações abaixo, as quais permitem a obtenção de um par de variáveis aleatórias com distribuição normal padronizada, a partir de dois números aleatórios R_1 e R_2 .

$$\begin{aligned} Z_1 &= \sqrt{-2 \ln R_1} \cos(2\pi R_2) \\ Z_2 &= \sqrt{-2 \ln R_1} \sin(2\pi R_2) \end{aligned}$$

Exemplo 9

Considerando as equações acima, gerar dois valores com distribuição normal padronizada a partir de $R_1 = 0,1758$ e $R_2 = 0,1489$.

$$Z_1 = [-2 \ln (0,1758)]^{\frac{1}{2}} \cos (2\pi 0,1489) = 1,11$$

$$Z_2 = [-2 \ln (0,1758)]^{\frac{1}{2}} \sin (2\pi 0,1489) = 1,50$$

Para a obtenção de variáveis aleatórias normais x_i , com média μ e desvio-padrão σ , deve-se aplicar a transformação $x_i = \mu + \sigma Z_i$ aos valores da normal padronizada. Por exemplo, para

transformar os valores obtidos em variáveis aleatórias normais com $\mu = 10$ e $\sigma = 2$, calcula-se:

$$x_1 = 10 + 2.(1,11) = 12,22$$

$$x_2 = 10 + 2.(1,50) = 13,00$$

1.5 Sumário

Neste anexo descreveu-se, primeiramente, a geração computacional de números aleatórios. Dentre os diversos métodos existentes, tratou-se com mais detalhes do Método Congruente Linear (MCL). Complementando esta introdução ao tema, apresentou-se alguns métodos de geração das principais variáveis aleatórias, contínuas e discretas. Todos eles dependem da geração de números aleatórios, os quais podem ser obtidos com os métodos vistos na primeira parte deste anexo.

Referências Bibliográficas

1. Banks, J. e Carson, J.S., *Discrete-Event System Simulation*, Prendice-Hall, Englewood Cliffs, NJ, 1984.
2. Banks, J. Carson, J.S. e Nelson, B.L., *Discrete-Event System Simulation*, 2nd ed., Prendice-Hall, Englewood Cliffs, NJ, 1996.
3. Law, A.M. e Kelton, W.D., *Simulation Modeling and Analysis*, 2nd ed., McGraw-Hill, NY, 1991
4. Bratley, P., Fox, B. L. e Schrage, L. E., *A Guide to Simulation*, 2nd. Ed., Springer-Verlag, NY, 1987.
5. Dudewicz E. J. e Karian Z. A., *Modern Design and Analysis of Discrete-Event Computer Simulations*, IEEE Computer Society Prees, 1985.
6. Freitas Filho, P.J., *Introdução à Modelagem e Simulação de Sistemas*, VisualBooks, 2008
7. Scharage, L. E., *A More Portable FORTRAN Randon Number Generator*, ACM Transactions on Mathematical Software, 5(2),132-138, 1979.
8. Jain, R., *The Art of Computer Systemns Performance Analysis*, Jhon Wiley & Sons, 1991.
9. Silva, V.L., Monte-Mor, J.A., Marcellino, F.J.M., Soma, N. Y., *Interface Amigável para o Diehard e Avaliação de Geradores de Números Pseudo-Aleatórios*, DCC-ITA, 2003