Geração de pontos aleatórios no R^n

Amarildo de Vicente* e Rogério Luiz Rizzi

*Centro de Ciências Exatas e Tecnológicas, Universidade Estadual do Oeste do Paraná, Rua Universitária, 2069, 85814-110, Cascavel, Paraná, Brasil. *Autor para correspondência. e-mail: amarildo@unioeste.br*

RESUMO. A geração de números aleatórios é um tema de grande interesse em virtude da sua importância em diversas áreas, como simulação, estatística, otimização por processos heurísticos, entre outras. Em particular, este trabalho foi desenvolvido em função de um problema que surgiu em uma pesquisa em que era necessário produzir pontos aleatórios em cubo do $\mathbf{R^n}$ (hipercubo) centrado na origem. A idéia inicial para produzir um ponto $X \in \mathbf{R^n}$ foi gerar n números reais aleatórios $x_1, x_2, ..., x_n$ uniformemente distribuídos no intervalo fechado [–L/2, L/2], onde L é o comprimento da aresta, e construir a n-upla $X = (x_1, x_2, ..., x_n)$. No entanto, percebeu-se que a grande maioria dos pontos produzidos dessa forma estava muito próxima da fronteira deste cubo, o que não era de interesse da pesquisa em questão. Este artigo visa então a esclarecer por que esse problema ocorre, bem como apresentar um algoritmo simples para solucioná-lo. Utilizando-se esse algoritmo, foram produzidas diversas seqüências de 4000 pontos cada uma, em cubos do $\mathbf{R^n}$, para diferentes valores de n. A análise desses dados mostrou que o algoritmo teve um bom desempenho, gerando pontos bem distribuídos nessas regiões.

Palavras-chave: números aleatórios, números pseudo-aleatórios, gerador de números aleatórios.

ABSTRACT. Random points generation in R^n. There is a large interest on random number generation for its importance in several areas as statistics, simulation, optimization by mean of heuristic process, etc. This paper in particular was developed based on a research where it was necessary to produce random points in a cube of $\mathbf{R^n}$. The initial idea to produce a random point was to generate n random real numbers $x_1, x_2, ..., x_n$ uniformly distributed in the [-L/2, L/2] interval, where L is the measure of the edges, and to build the point $X = (x_1, x_2, ..., x_n)$. However, it was noticed that most of the points produced on this manner were too near to the frontier of cube, what was not of interest to the research. This paper aims to show the reasons why this problem occurs and to present a single algorithm to solve it. Several sequences of 4000 points each in cubes of $\mathbf{R^n}$ were generated utilizing this algorithm, for different values of n. The analysis of data showed that the algorithm had a good performance, generating well distributed points in these regions.

Key words: random numbers, pseudo-random numbers, random number generator.

Introdução

A geração de números aleatórios é um processo bastante útil em diversos campos da Ciência, como simulação, otimização, probabilidade, etc. Por exemplo, em simulação, utiliza-se a geração de números aleatórios para simular a chegada de pessoas em uma fila, a fim de avaliar o tempo de espera; para simular a chegada de automóveis em um semáforo, com o propósito de avaliar a melhor forma de calibrá-lo, etc. Em otimização, pode-se utilizar tal processo em algoritmos genéticos, a fim de produzir indivíduos de uma população; no processo *Ant Colony Optimization*, a fim de gerar indivíduos na região de busca, etc.

Para gerar números aleatórios, há diversos geradores, geralmente contidos em pacotes de *softwares*, em calculadoras, em aplicativos como Excel, Maple e similares. Esses geradores, na verdade, são funções que geram números pseudo-aleatórios já que, a partir de um valor inicial (semente), geram uma seqüência fixa de números, como pode ser visto em L'ercuyer (1994), Ripley (1990) e Hellekalek (1998). Todavia, neste texto, tais números serão chamados simplesmente de aleatórios.

Um gerador de números aleatórios reais deve gerar tais números uniformemente distribuídos num intervalo I. Para o estudo pretendido neste trabalho, será considerado I = [0, 1]. A extensão desse intervalo para um intervalo J = [a, b] arbitrário pode ser feita facilmente por translação, contração ou

dilatação, já que sempre é possível construir uma função f bijetora de [0, 1] em [a, b]. Por exemplo, se $x \in [0, 1]$, então $f(x) = 10x$ leva x em [0, 10] (dilatação); $f(x) = 0.5x$ leva x em [0, 0.5] (contração); $f(x) = x + 1$ leva x em [1, 2] (translação).

Um bom gerador deve apresentar, pelo menos, as seguintes características: a) uniformidade dos pontos produzidos no intervalo especificado, b) independência estatística entre os números gerados, c) possibilidade de reprodução das seqüências geradas sem que essas sejam repetitivas para sementes diferentes e d) rapidez na geração de uma seqüência.

Um exemplo de gerador é o gerador linear congruente dado por

$X_{n+1} = (aX_n + b) \bmod m, n = 0, 1, 2, ..., m > 0, \quad (1)$

onde X_n é o termo geral da seqüência de números aleatórios, sendo a, b, e m constantes naturais e X_0 é um valor natural arbitrário (semente).

A expressão em (01) indica que X_{n+1} é o resto da divisão de $aX_n + b$ por m. Obviamente $X_{n+1} < m$.

As seqüências produzidas por esse gerador apresentam todas as características mencionadas anteriormente. Além disso, tais seqüências são cíclicas, isto é, a partir de uma dada semente, geram uma certa quantidade de números e, a partir de um determinado termo, os valores começam a se repetir. Por exemplo, para $a = 5$, $b = 1$, $m = 16$ e $X_0 = 1$, a seqüência produzida é $(X_n) = (1, 6, 15, 12, 13, 2, 11, 8, 9, 14, 7, 4, 5, 10, 3, 0, \textit{1, 6, 15,12, 13, 2, 11}, ...)$. Note-se que essa seqüência é formada por períodos, compostos pelos números 1, 6, 15, 12, 13, 2, 11, 8, 9, 14, 7, 4, 5, 10, 3 e 0.

Para que os valores gerados estejam no intervalo [0, 1] basta dividi-los por m (no caso 16) e, para que estes estejam em [0, 1], basta dividi-los por m–1 (15).

Conforme exposto anteriormente, a geração de pontos aleatórios uniformemente distribuídos no intervalo [0, 1] é uma tarefa que tem sido realizada com bastante sucesso. Todavia, no $\mathbf{R^n}$, é preciso tomar alguns cuidados para que tal distribuição seja realmente uniforme.

Geração de pontos no R^n

Neste texto, estaremos tratando distâncias sempre através da norma do máximo, definida a seguir.

Definição 1: Seja $X = (x_1, x_2, ..., x_n) \in \mathbf{R^n}$, chama-se norma do máximo, que se representa por $\| \ \|_\infty$, a norma dada por $\|X\|_\infty$ = máximo$\{|x_1|, |x_2|, ..., |x_n|\}$.

Considere um gerador G que produz pontos aleatórios uniformemente distribuídos no intervalo I = [0, 1] em um sistema decimal. Suponhamos que os números gerados por G possuam k casas decimais. Com essa precisão, o intervalo I fica dividido em 10^k partes iguais, de modo que o conjunto C de pontos gerados por G terá $10^k + 1$ elementos. Por exemplo, para k = 2 C possuirá 100 números x_i distintos da forma $x_i = 0.d_1d_2$, onde $0 \le d_1, d_2 \le 9$, mais o número 1, totalizando 101 elementos (10^2+1).

Consideremos o subintervalo [0.9, 1] de I. Evidentemente, qualquer número $x \in C$ tal que $x \ge 0.9$ é um número desse intervalo, distando, no máximo, 0.1 de 1 (supremo do intervalo I). Entre os elementos de C, há $10^{k-1} + 1$ deles com essa característica. Como em C há $10^k + 1$ elementos, então a probabilidade P de que um desses números ocorra ao se acionar o gerador G é $P = \dfrac{10^{k-1}+1}{10^k+1}$.

Para $k \ge 2$, nota-se que P tende a 0.1. Por exemplo, tomando-se os valores 2, 3 e 4 para k, obtêm-se aproximadamente os valores 0.1089, 0.1009 e 0.1001 para P, respectivamente. Fazendo-se k crescer indefinidamente, essa seqüência converge para 0.1, pois $\lim_{k\to\infty} \dfrac{10^{k-1}+1}{10^k+1} = \dfrac{1}{10}$. Isso significa que, em média, 10% dos números gerados por G estarão na faixa que vai de 0.9 a 1, incluindo os extremos. De um modo geral, para a precisão k considerada, dado $\varepsilon > 0$ a probabilidade P de que um número x gerado por G caia na faixa de $1 - \varepsilon$ até 1 será $P = \dfrac{\varepsilon 10^{k-1}+1}{10^k+1} \approx \dfrac{\varepsilon}{10}$. A prova disso, analogamente ao caso anterior, pode ser feita fazendo-se $\lim_{k\to\infty} \dfrac{\varepsilon 10^{k-1}+1}{10^k+1}$.

Particularizando-se a discussão anterior para o espaço $\mathbf{R^2}$, geometricamente um ponto $X = (x_1, x_2)$ em que pelo menos uma de suas coordenadas está na faixa de 0.90 a 1.00, em módulo, isto é, $0.90 \le |x_1| \le 1.00$ ou $0.90 \le |x_2| \le 1.00$, é um ponto que dista, no máximo, 0.1 da fronteira do quadrado de lados de medida 2, centrado na origem (ver Figura 1). Ao consideramos essa faixa, o referido quadrado fica dividido em duas regiões, as quais estão sendo chamadas de coroa e quadrado interno.

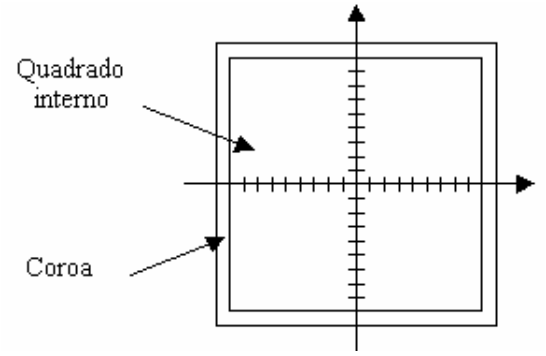

Figura 1. Quadrado com coroa.

Definição 2: Seja R uma região fechada do $\mathbf{R^n}$, dado um $\varepsilon > 0$, denomina-se coroa de R, com espessura ε, ao conjunto dos pontos $X \in R$ tais que $d(X, Fr) \leq \varepsilon$, onde d(X, Fr) é a distância de X até Fr e Fr é a fronteira externa do cubo. Por exemplo, X = (0.1, 0. 95) e Y = (0.5, –0.98) são pontos que estão na coroa da Figura 1.

Agora, imagine que se queira utilizar esse gerador para produzir um ponto aleatório $X = (x_1, x_2, ..., x_{10})$ no espaço $\mathbf{R^{10}}$, no interior do cubo de arestas 2, com centro na origem (similar ao quadrado da Figura 1). Então, para compor X, deverão ser gerados 10 números reais aleatórios $x_1, x_2, ..., x_{10}$, em [0, 1] e, em seguida, deverão ser acrescentados sinais + ou – a cada coordenada, de modo também aleatório. Conforme discutido anteriormente, é provável que pelo menos um desses números esteja na faixa de 0.90 a 1.00, em módulo. Isso quer dizer que o ponto X gerado deverá estar próximo à fronteira desse cubo, a uma distância máxima 0.1.

Muito embora no que foi exposto anteriormente tenham sido consideradas duas casas decimais para os números gerados por G, a rigor, tal consideração é desnecessária. Independentemente do número de casas decimais consideradas, sempre ocorrerá que 10% deles estarão na faixa de 0.90 a 1.00. De fato, para uma variável aleatória X com distribuição uniforme em [0, 1], sua função de densidade de probabilidade é $f(x) = 1$, se $0 \leq x \leq 1$ e 0 nos demais casos. Então a probabilidade P de que $0.90 \leq X \leq 1.00$ será dada por $P(0.90 \leq X \leq 1.00) = \int_{0.9}^{1.0} 1 dx =$ 0.1 (10%).

Nota 1: muito embora as nomenclaturas corretas para o $\mathbf{R^n}$ sejam hipercubo, hiperplano e hipervolume, continuaremos chamando esses elementos simplesmente de cubo, plano e volume, respectivamente.

Estudo do problema em um cubo

Consideremos o triângulo eqüilátero da Figura 2-a cujos lados medem L. Evidentemente seu perímetro é P1 = 3L. Suponhamos agora que, em cada um de seus lados, seja retirado um segmento de comprimento L/3 na parte central e, a seguir, sejam adicionados dois segmentos, ambos de comprimento L/3, conforme ilustra a Figura 2-b. Os 12 segmentos que compõem essa figura medem L/3, de modo que o perímetro da mesma será P2 = 12(L/3) = 3(4/3)L. Adotando-se o mesmo procedimento para a Figura 2-b, obtém-se a Figura 2-c, que contém 48 segmentos de comprimento L/9 e cujo perímetro é $P3 = 48(L/9) = 3(16/9)L = 3(4/3)^2L$.

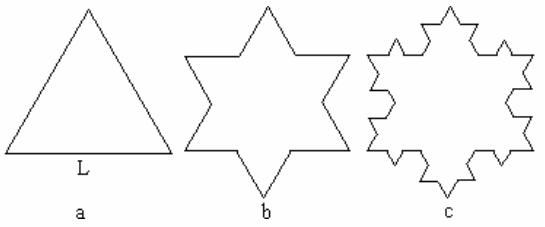

Figura 2. Construção do Floco de Neve de Koch.

Procedendo-se dessa forma, na n-ésima figura do processo, n = 0, 1, 2,..., o perímetro será $P = 3(4/3)^nL$. Essa figura é chamada de Floco de Neve de Koch (ver Serra, 1997). Agora, note-se que, se fizermos n tender a infinito, o perímetro tenderá a infinito, muito embora a área permaneça finita. Um comportamento similar pode ser observado em relação às áreas da coroa e do quadrado interno da Figura 1, conforme será descrito a seguir.

Consideremos então o quadrado com lados de comprimento 2 centrado na origem. Sejam os números gerados pelo gerador G citado anteriormente. Continuemos a admitir que, para nossos propósitos, os números gerados na faixa de 0.90 a 1.00 (em módulo) são números próximos à fronteira. Para o quadrado mencionado, essa faixa produz uma coroa de espessura 0.1 (ver Figura 1).

Notemos agora que a área AQ do quadrado interno vale 1.8^2, ao passo que a área AC da coroa vale $4 – 1.8^2$. Assim, a razão de AC para AQ é $r_Q = (4 – 1.8^2)/1.8^2 \cong 0.23$.

Sob as mesmas condições, consideremos no R^3 um cubo com arestas de comprimento 2, centrado na origem. O volume do cubo interno é $VCB = 1.8^3$, ao passo que o volume da coroa é $VCR = 8 – 1.8^3$. Nesse caso, a razão do volume da coroa externa para o volume do cubo interno é $r_C = (8 – 1.8^3)/1.8^3 \cong 0.37$.

Percebe-se então que $r_C > r_Q$ e que, para o cubo, o volume da coroa é de aproximadamente 37% do

volume da parte interna (cubo menor), ao passo que, para o quadrado, a área da coroa representa aproximadamente 23% da área da parte interna (quadrado interno). Pode-se mostrar que, com uma espessura fixada ε para a coroa, à medida que a dimensão do espaço aumenta, o volume dessa coroa tende a aumentar indefinidamente, superando o volume da parte interna do cubo.

Teorema 1: Seja C_L um cubo de arestas L contido no $\mathbf{R^n}$ e uma coroa de espessura $\varepsilon > 0$ para C_L. Sendo VCB o volume do cubo e VCR o volume da coroa, então $\lim_{n\to\infty} \frac{VCB}{VCR} = \infty$.

De fato, considere um cubo de arestas L no $\mathbf{R^n}$. Imaginemos uma espessura $\varepsilon > 0$ para a coroa desse cubo. Então o volume dessa coroa será $L^n - (L - 2\varepsilon)^n$, ao passo que o volume do interior será $(L - 2\varepsilon)^n$. Desse modo, a razão do volume da coroa para o volume do interior será

$$r = \frac{L^n - (L-2\varepsilon)^n}{(L-2\varepsilon)^n} = \frac{L^n}{(L-2\varepsilon)^n} - \frac{(L-2\varepsilon)^n}{(L-2\varepsilon)^n} = \frac{L^n}{(L-2\varepsilon)^n} - 1 = \left(\frac{L}{L-2\varepsilon}\right)^n - 1.$$

Notando que $\frac{L}{L-2\varepsilon} > 1$, então $\lim_{n\to\infty}\left(\frac{L}{L-2\varepsilon}\right)^n = \infty$, de modo que $r \to \infty$. Isso significa que, para n grande, o volume da coroa é maior que o volume do interior do cubo e, por essa razão, tende a atrair os pontos gerados por G. Esse resultado, associado ao que foi exposto no item 2, indica que a maioria dos pontos produzidos por G cairá próxima à fronteira do cubo. Desse modo, gerar um ponto aleatório no interior de uma região do $\mathbf{R^n}$ não é uma tarefa que se realiza simplesmente gerando n números reais (coordenadas) de forma aleatória, uniformemente distribuídos num intervalo [a, b].

Até aqui foi considerada uma faixa de interesse próxima à fronteira da região em questão. Todavia, a coroa mencionada ocorre automaticamente sempre que se trabalha com números de ponto flutuante.

De fato, suponhamos que o gerador G gere números com k casas decimais no intervalo [0, 1]. Então o primeiro número menor que 1 possível de ser produzido é $r = 0.\underbrace{999...9}_{k\ \text{dígitos}}$. Com o arredondamento, se um número x for produzido em [0.99...9, 1], então esse será arredondado para 1, se x $\geq \frac{r+1}{2} = 0.\underbrace{999...95}_{k+1\ \text{dígitos}}$, ou para r, se x < $\frac{r+1}{2}$ (ver Figura 3).

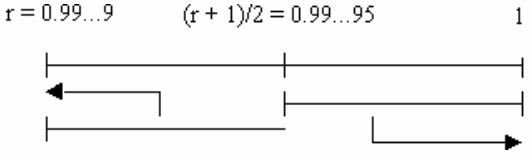

Figura 3. Esquema de arredondamento de um número.

Desse modo, um número gerado na faixa de (r + 1)/2 a 1, incluindo os extremos, cairá sempre na fronteira da região, ou seja, será arredondado para 1. Isso significa que a fronteira na verdade não é um plano, mas sim uma coroa de espessura $\varepsilon = 1 - 0.\underbrace{999...95}_{k+1\ \text{dígitos}} = 0.510^{-k}$.

Método para gerar ponto aleatórios em um cubo do R^n

Considere $C(L, X_0)$ um cubo contido no $\mathbf{R^n}$, com semi-arestas de medida L e com centro num ponto $X_0 = (x_1, x_2, ...x_n)$. Conforme já exposto, um ponto aleatório X desse cubo em que suas coordenadas são produzidas de maneira uniformemente distribuídas no intervalo [0, L] está quase sempre próximo da fronteira do mesmo. Para que os pontos produzidos estejam no fecho de tal cubo (união da fronteira com o interior), pode-se empregar o método a seguir.

Divida o cubo em m coroas c_1, c_2, ..., c_m de amplitude h = L/m (ver Figura 4). Escolha aleatoriamente uma coroa c_i. Gere um ponto aleatório X' em c_i, de modo que suas coordenadas sejam produzidas aleatória e uniformemente distribuídas no intervalo [0, ih].

O ponto X' provavelmente estará próximo da fronteira do cubo C(O, ih), sendo O a origem. A seguir, deve-se fazer a translação de X' para o ponto $X = X' + X_0$, translação do ponto X' do cubo C(O, ih) para o cubo $C(X_0, ih)$. O ponto X produzido será então um ponto aleatório de $C(X_0, L)$. Esse processo pode ser realizado através do algoritmo a seguir, sendo que G(r) representa um gerador que produz números aleatórios reais em [0, r] e GI(n) representa um gerador que produz números aleatórios inteiros em [1, n].

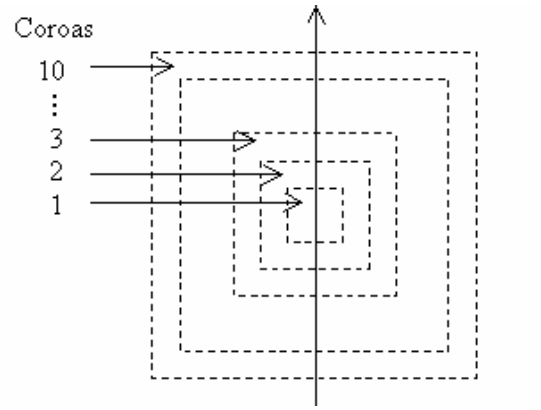

Figura 4. Coroas particionando um quadrado centrado na origem, para o caso em que m = 10.

Algoritmo para gerar um ponto aleatório num cubo do R^n

Entrada: X_0 (centro do cubo), m (número de coroas) e L (semi-aresta do cubo);
1. h := L/m;
2. i := GI(m) (seleção aleatória de uma coroa c_i, i = 1, 2, ..., m);
3. Repita
4. Gere as n coordenadas x_j de X fazendo x_j:= ih[2G(1) – 1];
5. Até que $i(h – 1) \leq ||X||_\infty \leq ih$;
6. X := X + X_0;
7. fim.

Vale ressaltar que o algoritmo acima produz pontos em todo o cubo C(X_0, L), mas não garante a distribuição uniforme desses pontos.

Nota 2: a norma tratada até aqui tem sido a norma do máximo, porém nada impede que se utilize outra norma qualquer.

Resultados

Para exemplificar o funcionamento do algoritmo anterior, são apresentadas, a seguir (Tabela 1), as etapas para a geração de dois pontos do $\mathbf{R}^3$. Neste exemplo, considerou-se o cubo C(X_0, 2) do $\mathbf{R}^3$, onde X_0 = (1, 1, 1). O número de coroas adotado foi m = 10, de modo que h = 0.2.

Tabela 1. Exemplo de pontos produzidos em um cubo do $\mathbf{R}^3$.

k	i = G(m)	x_1 = ih[2G(1)–1]	x_2 = ih[2G(1)–1]	x_3 = ih[2G(1)–1]	Pontos gerados
1	10	2[2(0.78)–1]	2[2(0.53)–1]	2[2(0.96)–1]	(1.12, 0.12, 1.84)
2	6	1.2[2(0.59)–1]	1.2[2(0.97)–1]	1.2[2(0.51)–1]	(0.22, 1.13, 0.02)

Após a translação X_k + X_0, os pontos produzidos são X_1 = (2.12, 1.12, 2.84) e X_2 = (1.22, 2.13, 1.02).

A seguir, são apresentadas duas tabelas (Tabelas 2 e 3) onde foram produzidos dez pontos aleatórios no cubo C(O, 1), onde O é a origem. No primeiro caso, cada coordenada foi produzida de modo aleatório no intervalo [0, 1], usando o processo padrão. No segundo caso, foi empregado o algoritmo proposto para produzir um ponto aleatório em um cubo do $\mathbf{R}^n$. Em ambos os casos, os números aleatórios foram produzidos pelo gerador do Pascal.

Tabela 2. Pontos produzidos pelo processo padrão.

k	Pontos X_k gerados	Distância de X_k até a fronteira
1	(0.25, 0.85, 0.08, 0.25, 0.56, 0.06, 0.93, 0.35, 0.91, 0.86)	0.07
2	(0.75, 0.07, 0.15, 0.43, 0.78, 0.28, 0.02, 0.42, 0.16, 0.65)	0.22
3	(0.53, 0.02, 0.82, 0.58, 0.78, 0.72, 0.13, 0.21, 0.04, 0.83)	0.17
4	(0.10, 0.47, 0.46, 0.84, 0.77, 0.65, 0.77, 0.98, 0.98, 0.82)	0.02
5	(0.83, 0.87, 0.01, 0.05, 0.94, 0.92, 0.41, 0.50, 0.12, 0.83)	0.04
6	(0.18, 0.45, 0.95, 0.55, 0.26, 0.15, 0.57, 0.36, 0.61, 0.98)	0.02
7	(0.94, 0.05, 0.75, 0.18, 0.54, 0.28, 0.31, 0.30, 0.96, 0.62)	0.04
8	(0.93, 0.06, 0.20, 0.93, 0.74, 0.29, 0.62, 0.96, 0.15, 0.03)	0.04
9	(0.74, 0.36, 0.61, 0.17, 0.41, 0.59, 0.70, 0.03, 0.15, 0.25)	0.26
10	(0.73, 0.83, 0.48, 0.97, 0.47, 0.96, 0.63, 0.76, 0.00, 0.49)	0.03

Tabela 3. Pontos produzidos através do algoritmo proposto.

k	Pontos X_k gerados	Distância de X_k até a fronteira
1	(0.52, 0.37, 0.30, 0.37, 0.08, 0.26 0.23, 0.52, 0.69, 0.43)	0.31
2	(0.12, 0.49, 0.54, 0.51, 0.54, 0.02 0.43, 0.10, 0.24, 0.50)	0.56
3	(0.25, 0.39, 0.38, 0.01, 0.16, 0.33 0.27, 0.35, 0.27, 0.39)	0.61
4	(0.02, 0.05, 0.00, 0.01, 0.05, 0.01 0.00, 0.01, 0.00, 0.01)	0.95
5	(0.08, 0.90, 0.69, 0.83, 0.47, 0.92 0.62, 0.28, 0.90, 0.45)	0.08
6	(0.09, 0.12, 0.26, 0.05, 0.22, 0.11 0.05, 0.22, 0.04, 0.06)	0.74
7	(0.30, 0.34, 0.04, 0.20, 0.08, 0.03 0.24, 0.12, 0.35, 0.31)	0.65
8	(0.15, 0.09, 0.22, 0.10, 0.29, 0.10 0.16, 0.28, 0.23, 0.15)	0.71
9	(0.32, 0.48, 0.21, 0.48, 0.07, 0.59 0.17, 0.44, 0.44, 0.55)	0.41
10	(0.56, 0.34, 0.15 ,0.26, 0.16, 0.33 0.43, 0.19, 0.04, 0.54)	0.44

Discussão e conclusão

Como pode ser percebido, na Tabela 2, quase todos os pontos estão muito próximos da fronteira do cubo. Já na Tabela 3, as distâncias são bastante variadas, comprovando que os pontos estão bem distribuídos (ver Figuras 5-a e 5-b).

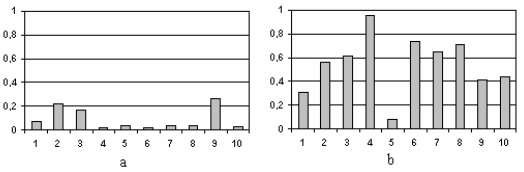

Figura 5. a) Gráfico referente à Tabela 2; b) Gráfico referente à Tabela 3.

Para uma análise mais ampla, foram produzidas diversas seqüências de pontos em cubos C(O, 10) do $\mathbf{R}^n$, cada uma com 4000 elementos. Esse experimento foi feito tanto para o processo padrão (geração de n coordenadas aleatórias no intervalo

[-10, 10] para cada ponto) quanto pelo algoritmo proposto. Em ambos os processos, foi mantida a mesma semente para o gerador de números aleatórios, sendo tal semente a dimensão do espaço $\mathbf{R}^n$ considerado. Esses cubos foram divididos em 10 coroas (m = 10), a começar pela origem, de modo que, para cada coroa, ficou fixada uma espessura h = 0.2 (ver Figura 4 para o caso do $\mathbf{R}^2$). Para cada uma dessas coroas, foi feita a contagem dos pontos ali gerados, considerando-se que um ponto gerado sobre a fronteira de uma coroa pertence à coroa que está à sua esquerda. Os resultados obtidos estão apresentados nas Tabelas 4 e 5.

Tabela 4. Distribuição dos pontos produzidos pelo processo padrão nas dez coroas consideradas.

Dimensão	Pontos por coroa									
2	53	127	190	293	380	432	529	564	671	761
4	0	2	28	90	171	285	401	674	1048	1301
6	0	1	7	14	31	135	261	556	1076	1919
8	0	0	0	2	15	45	175	445	1040	2278
10	0	0	0	0	0	18	88	340	939	2615
40	0	0	0	0	0	0	0	0	6	3937
60	0	0	0	0	0	0	0	0	3	3997
80	0	0	0	0	0	0	0	0	3	3997
100	0	0	0	0	0	0	0	0	0	4000

Tabela 5. Distribuição dos pontos produzidos pelo algoritmo proposto nas dez coroas consideradas.

Dimensão	Pontos por coroa									
2	397	360	407	385	435	430	366	427	396	397
4	388	396	415	403	401	374	372	459	393	399
6	394	385	406	402	433	407	391	409	360	413
8	389	397	395	403	385	426	394	383	422	406
10	421	420	402	349	411	388	390	389	416	414
20	409	386	427	370	409	383	429	416	417	354
40	382	424	409	379	415	374	395	417	405	400
60	395	412	362	398	456	384	374	395	420	404
80	364	388	368	416	401	411	402	435	376	439
100	396	411	363	397	456	384	376	393	422	402

Como fica claro na Tabela 5, os pontos produzidos pelo algoritmo proposto se distribuem de maneira uniforme nas dez coroas consideradas, para todas as dimensões do $\mathbf{R}^n$ tratadas. Já o processo padrão, em que cada ponto é produzido gerando-se n coordenadas aleatórias no intervalo [-10, 10], faz que os pontos se dirijam para a fronteira do cubo à medida que a dimensão aumenta. Esses resultados indicam que, para os propósitos deste trabalho, o algoritmo apresentado teve um bom desempenho.

Referências

HELLEKALEK, P. Good random number generators are (not so) easy to find. *Mathematics and Computers in Simulation,* Amsterdam, v. 46, p. 485-505, 1998.

L'ECUYER, P. Uniform random number generation. *Operational Research Quaterly,* London, v.53, n.1, p. 77-120, 1994.

RIPLEY, B. D. Thoughts on pseudorandom number generators. *Journal Computational Applied Mathematical,* v.31, n.1, p. 153-163, 1990.

SERRA, C. P. *et al*. *Fractais gerados por sistemas dinâmicos*. Curitiba: Champagnat, 1997.

Received on June 24, 2003.
Accepted on November 27, 2003.