Introdução a Métodos Computacionais em Física
Módulo 5

Leonardo Cabral

19 de Setembro de 2019

UFPE

DF-UFPE

Objetivos

- Estudar como modelar processos aletórios numericamente.
  - (Pseudo-)geradores de números aleatórios.
  - Caminhantes aleatórios.
  - Processo de Monte Carlo.
- Estudar a conexão entre caminhantes aleatórios e a equação de difusão.

Mais informações sobre este assunto podem ser encontradas no Cap. 7 do H. Gould, J. Tobochnik e W. Christian, "An Introduction to Computer Simulations Methods: Applications to Physical systems", 3^{rd} Ed.

Sequências de números pseudo-aleatórios

- Sequências de números aleatórios podem ser produzidas por eventos físicos
- Na prática, sequências de números pseudo-aleatórios podem ser construídas numericamente. Possuem um período, a partir do qual a sequência se repete.
- Diferença entre pseudo-sequências e sequências de números verdadeiramente aleatórios é irrelevante desde que o gerador de números pseudo-aleatório seja:
  - eficiente;
  - independa da máquina;
  - reprodutível;
  - e satisfaça testes estatísticos específicos.
- Gerador de números aleatórios mais utilizado é o *linear congruential method*. É rápido e consiste no mapa

$$x_{n+1} = (ax_n + c) \bmod m, \qquad a,\, c,\, m\, \{x_n\} \in \mathbb{Z}$$

a partir de uma semente x_0. Período máximo deste gerador é igual a m.

Sequências de números pseudo-aleatórios

- Exemplos de sequência gerada pelo *linear congruential method* para $m = 32$, $a = 3$, $c = 4$ e semente:

$$
\begin{aligned}
x_0 = 0 \rightarrow &\quad \{4,\ 16,\ 20,\ 0,\ 4, \ldots\} \\
x_0 = 1 \rightarrow &\quad \{7,\ 25,\ 15,\ 17,\ 23,\ 9,\ 31,\ 1,\ 7, \ldots\} \\
x_0 = 2 \rightarrow &\quad \{10,\ 2,\ 10, \ldots\} \\
x_0 = 3 \rightarrow &\quad \{13,\ 11,\ 5,\ 19,\ 29,\ 27,\ 21,\ 3,\ 13, \ldots\} \\
x_0 = 6 \rightarrow &\quad \{6,\ 22,\ 6, \ldots\} \\
x_0 = 8 \rightarrow &\quad \{28,\ 24,\ 12,\ 8,\ 28, \ldots\} \\
x_0 = 14 \rightarrow &\quad \{14,\ 14, \ldots\} \\
x_0 = 18 \rightarrow &\quad \{26,\ 18,\ 26 \ldots\} \\
x_0 = 30 \rightarrow &\quad \{30,\ 30, \ldots\}
\end{aligned}
$$

- Para o mesmo $m = 32$, outros exemplos podem ser gerados utilizando diferentes valores de a e c que possuam períodos mais longos (e.g., $a = 3$, $c = 5$ e $a = 5$, $c = 7$).

UFPE

DF-UFPE

Generalized feedback shift register method

- Utiliza manipulação de bits para gerar sequência de números pseudo-aleatórios e operador `exclusive or` (xor) ou $\oplus$. Em `C` é denotado por ˆ,

$$x_n = x_{n-p} \oplus x_{n-q}, \qquad x_n,\, p,\, q \in \mathbb{Z},$$

e $p > q$. Por exemplo, sejam os primeiros 5 números da sequência $\{7,\, 25,\, 15,\, 17,\, 23\}$. Se $p = 5$, $q = 3$, então

$$x_6 = x_1 \oplus x_3 = 7 \oplus 15 = \left(\begin{array}{cc} & 00000111 \\ \oplus & 00001111 \\ = & 00001000 \end{array}\right) = 8;$$

$$x_7 = x_2 \oplus x_4 = 25 \oplus 17 = \left(\begin{array}{cc} & 00011001 \\ \oplus & 00010001 \\ = & 00001000 \end{array}\right) = 8$$

$$x_8 = x_3 \oplus x_5 = 15 \oplus 23 = \left(\begin{array}{cc} & 00001111 \\ \oplus & 00010111 \\ = & 00011000 \end{array}\right) = 24$$

$$x_9 = x_4 \oplus x_6 = 17 \oplus 8 = \left(\begin{array}{cc} & 00010001 \\ \oplus & 00001000 \\ = & 00011001 \end{array}\right) = 25$$

$$x_{10} = x_5 \oplus x_7 = 23 \oplus 8 = \left(\begin{array}{cc} & 00010111 \\ \oplus & 00001000 \\ = & 00011111 \end{array}\right) = 31$$

$$x_{11} = x_6 \oplus x_8 = 8 \oplus 24 = \left(\begin{array}{cc} & 00001000 \\ \oplus & 00011000 \\ = & 00010000 \end{array}\right) = 16$$

Gerador rand()

```c
#include <stdlib.h>


/* This algorithm is mentioned in the ISO C standard, here extended
   for 32 bits.  */
int
rand_r (unsigned int *seed)
{
  unsigned int next = *seed;
  int result;

  next *= 1103515245;
  next += 12345;
  result = (unsigned int) (next / 65536) % 2048;

  next *= 1103515245;
  next += 12345;
  result <<= 10;
  result ^= (unsigned int) (next / 65536) % 1024;

  next *= 1103515245;
  next += 12345;
  result <<= 10;
  result ^= (unsigned int) (next / 65536) % 1024;

  *seed = next;

  return result;
}
```

Na prática, o que fazer?

- Há muito mais do que `rand()` do `C/C++`!
- `C++11` possui implementações de diversos algoritmos para gerar sequências de números aleatórios, inclusive com distribuições não-uniformes. Ver: http://en.cppreference.com/w/cpp/numeric/random
- Se seu programa requer números aleatórios somente para produzir resultados diferentes, sem ser necessário que o número produzido seja de uma boa sequência de números aleatórios, `rand()` vai satisfazer suas necessidades.
- Por outro lado, se for necessária a geração de números aleatórios para produzir sequências que simulem flutuações térmicas, por exemplo, então é preciso utilizar um bom gerador de números aleatórios. Deve-se utilizar gerador melhor do que `rand()`.

UFPE

DF-UFPE

Testes

- Período
- Uniformidade
- Teste χ^2
- Preenchimento de sítios
- Teste do 'estacionamento'
- Correlações escondidas
- Correlações de curto alcance
- Caminhante aleatório

Probabilidades: p ($\rightarrow$) e $q = 1 - p$ ($\leftarrow$), passo $s_i = \pm a$.

- Após N passos, $x_N = \sum_{i=1}^{N} s_i$ da origem.
- Média de x_N após N passos:

$$\langle x_N \rangle = \sum_{i=1}^{N} \langle s_i \rangle = N \langle s \rangle = N \left[p\, a + q\,(-a) \right] = (p - q) N a$$

- Desvio quadrático médio após N passos:

$$\Delta x_N^2 = \langle (x_N - \langle x_N \rangle)^2 \rangle = \left\langle \sum_{i=1}^{N} \delta_i \sum_{j=1}^{N} \delta_j \right\rangle = \sum_{i=1}^{N} \langle \delta_i^2 \rangle + \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} \langle \delta_i \delta_j \rangle = 4pqNa^2$$

$$\langle \delta_i \delta_j \rangle = \langle \delta_i \rangle \langle \delta_j \rangle = \langle s_i - \langle s \rangle \rangle \langle s_j - \langle s \rangle \rangle = \left[N(\langle s \rangle - \langle s \rangle) \right]^2 = 0 \qquad \text{passos independentes}$$

$$\langle \delta_i^2 \rangle = \langle s_i^2 \rangle - \langle s \rangle^2 = \left(1 - (p - q)^2 \right) a^2 = \left((p + q)^2 - (p - q)^2 \right) a^2 = 4pqNa^2$$

Procedimento para gerar uma caminhada aleatória

1. Para gerar uma dada caminhada aleatória numericamente, pode-se utilizar um processo de Monte-Carlo.
2. Seja p a probabilidade do caminhante ir para a direita. Em cada passo da caminhada, gere um número aleatório, r. Se

$$\begin{aligned} p < r \text{ for verdadeiro,} \qquad & \text{vá para a direita} \\ p < r \text{ for falso,} \qquad & \text{vá para a esquerda} \end{aligned}$$

Em suma, este algoritmo é um processo de Monte-Carlo aplicado à caminhada aleatória.

Probabilidades: p ($\rightarrow$) e $q = 1 - p$ ($\leftarrow$), passo $s_i = \pm a$.

1. Inicie com o caminhante na origem. Em cada passo, um gerador de número aleatório produz um número r tal que:

$$\text{Se } p < r, \text{ vá para } \rightarrow; \qquad \text{Se } p \geq r, \text{ vá para } \leftarrow$$

2. Escreva um programa que implemente o caminhante aleatório em 1D. Para cada valor de p realize M tentativas (ou implemente M caminhantes independentes). Durante a execução do programa calcule $\langle x \rangle$ e $\langle x^2 \rangle$ para números de passos N diferentes. Utilize $M = 10$, $M = 100$ e $M = 10000$. O que acontece com seus resultados quando M aumenta?

3. A distribuição de probabilidade $P(x)$ de x_n estar no valor x no passo n pode ser obtida a partir do histograma dos valores para x_i, $i = 1, 2, \ldots, N$, para $M > 1000$ caminhantes diferentes. Compare seus resultados com

$$P_N(x) = \frac{C}{\sqrt{2\pi\Delta x^2}} \exp\left(-\frac{(x - \langle x \rangle)^2}{2\Delta x^2}\right).$$

Qual é o valor de C para o qual se obtem bom ajuste?

UFPE

DF-UFPE

Caminhante aleatório em uma rede quadrada:

4. Considere uma coleção de $M > 1000$ caminhantes aleatórios independentes inicialmente na origem em 2D.
5. Em cada passo de tempo cada caminhante pode ir para qualquer uma das quatro direções possíveis com igual probabilidade ($p = 1/4$).
6. Utilize a biblioteca `OpenGL` para visualizar a posição de cada caminhante em diferentes passos de tempo. Realize no mínimo 500 passos de tempo. Como a figura obtida evolui com o tempo.
7. Calcule $\langle x \rangle$, $\langle x^2 \rangle$, $\langle y \rangle$, $\langle y^2 \rangle$ e o deslocamento médio quadrático

   $$R^2 = \langle x^2 \rangle - \langle x \rangle^2 + \langle y^2 \rangle - \langle y \rangle^2,$$

   (as médias devem ser calculadas sobre os M caminhantes) como função do número de passos, N.
8. Estime R^2 para $N = 8,\ 16,\ 32,\ 64,\ \ldots,$, calculando as médias para um número grande de caminhantes para cada valor de N. Salve os valores obtidos e, em gráfico log-log, obtenha o valor de ν da dependência,

   $$R \sim N^{\nu}, \qquad (N \gg 1)$$

   Se você obtiver $\nu = 1/2$ estime a magnitude do coeficiente de auto-difusão D a partir de $R^2 = 4DN$.

Caminhante aleatório em uma rede triangular:

8. Refaça os itens para um caminhante em uma rede quadrada para o caso de um caminhante aleatório percorrendo uma rede triangular, onde as probabilidades de ir nas direções $\hat{x}$, $-\hat{x}$, $\cos\pi/2\hat{x} + \operatorname{sen}\pi/2\hat{y}$, $-\cos\pi/2\hat{x} + \operatorname{sen}\pi/2\hat{y}$, $-\cos\pi/2\hat{x} - \operatorname{sen}\pi/2\hat{y}$ e $\cos\pi/2\hat{x} - \operatorname{sen}\pi/2\hat{y}$ são todas iguais a $1/6$.
9. Tente implementar o caminhante aleatório em 2D no contínuo (Problema 7.13 do Tobochnik). Neste caso, considere um passo de tamanho fixo, mas cuja orientação é aleatória. Refaça os itens anteriores.

Equação de difusão:

$$\frac{\partial P}{\partial t} = D\frac{\partial^2 P}{\partial x^2} - v\frac{\partial P}{\partial x}$$

onde $P(x,\, t)$ é a distribuição de probabilidade, D o coeficiente de difusão e v, velocidade. Tem-se,

$$\int_{-\infty}^{\infty} x\frac{\partial P}{\partial t}dx = D\int_{-\infty}^{\infty} x\frac{\partial^2 P}{\partial x^2}dx - v\int_{-\infty}^{\infty} x\frac{\partial P}{\partial x}dx \quad \rightarrow \quad \frac{d\langle x\rangle}{dt} = v$$

$$\int_{-\infty}^{\infty} x^2\frac{\partial P}{\partial t}dx = D\int_{-\infty}^{\infty} x^2\frac{\partial^2 P}{\partial x^2}dx - v\int_{-\infty}^{\infty} x^2\frac{\partial P}{\partial x}dx \quad \rightarrow \quad \frac{d\langle x^2\rangle}{dt} = 2\left(D + v\langle x\rangle\right)$$

onde $\lim_{|x|\to\infty} x^n\frac{\partial^m}{\partial x^m}P(x,\, t) \to 0$, para $n,\, m = 0,\, 1,\, 2,\, \ldots$.

Logo, se $v = \text{const.}$, $\langle x\rangle = vt$ e $\Delta x^2 = 2Dt$.

Caminhante aleatório: equação mestre

$$P_i^{(n)} = pP_{i-1}^{(n-1)} + qP_{i+1}^{(n-1)}$$

onde $P_i^{(n)}$ é a probabilidade do caminhante estar no sítio i depois de n passos. Se cada passo é realizado em um intervalo δt e os sítios estão separados por δx,

$$P_i^{(n)} = \int_{x-\delta x/2}^{x-\delta x/2} d\eta P(\eta, t) = \delta x P(x,\ t)$$

$$P(x,\ t) = pP(x - \delta x,\ t - \delta t) + qP(x + \delta x,\ t - \delta t)$$

$$\frac{P(x,\ t) - P(x,\ t - \delta t)}{\delta t} = \frac{P(x - \delta x,\ t - \delta t) + P(x + \delta x,\ t - \delta t) - 2P(x,\ t - \delta t)}{2\delta t}$$
$$+ \frac{\left(p - \frac{1}{2}\right) P(x - \delta x,\ t - \delta t) + \left(q - \frac{1}{2}\right) P(x + \delta x,\ t - \delta t)}{\delta t}$$

Mas

$$P(x \pm \delta x,\ t) = P(x,\ t) \pm \delta x \frac{\partial}{\partial x} P(x,\ t) + \frac{1}{2}\delta x^2 \frac{\partial^2}{\partial x^2} P(x,\ t) + \mathcal{O}(\delta x^3)$$

Caminhante aleatório (cont)

$$\frac{P(x-\delta x,\,t-\delta t)+P(x+\delta x,\,t-\delta t)-2P(x,\,t-\delta t)}{2\delta t}=\frac{\delta x^2}{2\delta t}\frac{\partial^2}{\partial x^2}P(x,\,t-\delta t)+\mathcal{O}(\delta x^3)$$

$$\frac{\left(p-\frac{1}{2}\right)P(x-\delta x,\,t-\delta t)+\left(q-\frac{1}{2}\right)P(x+\delta x,\,t-\delta t)}{\delta t}=\frac{(p+q-1)}{\delta t}\left(P(x,\,t-\delta t)+\frac{1}{2}\delta x^2\frac{\partial^2}{\partial x^2}P(x,\,t-\delta t)\right)-$$

$$-\frac{(p-q)\,\delta x}{\delta t}\frac{\partial}{\partial x}P(x,\,t-\delta t)+\mathcal{O}(\delta x^3)=-\frac{(p-q)\,\delta x}{\delta t}\frac{\partial}{\partial x}P(x,\,t-\delta t)+\mathcal{O}(\delta x^3)$$

Portanto,

$$\frac{P(x,\,t)-P(x,\,t-\delta t)}{\delta t}=\frac{\delta x^2}{2\delta t}\frac{\partial^2}{\partial x^2}P(x,\,t-\delta t)-\frac{(p-q)\,\delta x}{\delta t}\frac{\partial}{\partial x}P(x,\,t-\delta t)+\mathcal{O}(\delta x^3)$$

Para $\delta x\to 0$ e $\delta t\to 0$,

$$\frac{\partial}{\partial t}P(x,\,t)=D\frac{\partial^2}{\partial x^2}P(x,\,t)-v\frac{\partial}{\partial x}P(x,\,t),$$

onde $D=\delta x^2/2\delta t$ e $v=(p-q)\,\delta x/\delta t$.