Centro Federal de Educação Tecnológica de Minas Gerais - CEFET-MG
Disciplina: Princípio de Modelagem Matemática
Dia/Horário: Segunda e Quarta-feira, das 14h50min às 16h30min

Random Walk

RELATÓRIO – PRÁTICA 02

Michelle Hanne Soares de Andrade

Belo Horizonte
Abril, 2023.

SUMÁRIO

1- INTRODUÇÃO

O método Random Walk (ou caminhada aleatória) é um modelo matemático utilizado para descrever sistemas dinâmicos que evoluem de forma probabilística. Esses sistemas são compostos por uma série de passos aleatórios em que a direção e a magnitude de cada passo são determinadas por uma distribuição de probabilidade (DYM e HUPPENREUSS, 2004).

O modelo Random Walk é utilizado em uma ampla gama de aplicações, incluindo finanças, biologia, física, química, entre outras áreas.

Para implementar o modelo de caminhadas aleatórias utilizou-se o método Monte Carlo. O método de Monte Carlo consiste em gerar sequências de números aleatórios para determinar os passos da partícula em cada iteração do processo. O gerador de números aleatórios congruencial é um algoritmo que produz uma sequência pseudoaleatória de números, baseando-se em uma relação de recorrência matemática.

Visando simular este problema, foi proposto um algoritmo na linguagem Pyhton que implementa um gerador de números aleatórios congruencial para o Random walk (posição x e tempo) de 10 caminhadas com 10.000 passos cada uma. Posteriormente, foram realizados o cálculo de desvio quadrático médio e o ajuste da curva com a lei de potência. Buscando avaliar o resultado foi calculado o teste de hipóteses com o valor obtido do expoente da lei de potência, mostrando que o caminhante execute um verdadeiro Random Walk.

E, por fim, foi calculado o coeficiente de Hurst, afim de verificar se os movimentos aleatórios gerados pelo modelo exibem dependência temporal.

1.1 - Objetivo

Reproduzir um algoritmo Random Walk implementado com o método de Monte Carlo, utilizando um gerador de números aleatórios congruencial. Assim, avaliar se é possível executar um verdadeiro Random Walk, no qual o expoente da lei de potência será igual a 1.

2- REVISÃO BIBLIOGRÁFICA

Caminhadas aleatórias são processos estocásticos que descrevem a evolução de uma partícula ou sistema em que cada passo é determinado aleatoriamente (DYM e HUPPENREUSS, 2004).

Um dos métodos para gerar números aleatórios é através de modo congruencial. O gerador de números aleatórios congruencial é um método simples e amplamente utilizado para gerar sequências de números pseudoaleatórios. É baseado em uma equação de recorrência que produz uma sequência de números inteiros entre 0 e 1. O algoritmo é caracterizado pela equação:

$$X_{n+1} = (a * X_n + c) \, mod \, m$$

onde X_n é o n-ésimo número da sequência, $a, c, e \, m$ são constantes escolhidas de maneira adequada para produzir uma sequência com propriedades de aleatoriedade (DYM e HUPPENREUSS, 2004).

O método de Monte Carlo é uma técnica de simulação estatística que utiliza sequências de números aleatórios para modelar fenômenos complexos e obter soluções aproximadas de problemas. Por exemplo, podemos simular uma caminhada aleatória em uma dimensão, em que a partícula começa na posição zero e pode se mover uma unidade para a direita ou para a esquerda em cada passo. Para cada iteração do processo, podemos gerar um número aleatório entre zero e um e, se esse número for menor que 0,5, a partícula se move uma unidade para a esquerda, caso contrário, ela se move uma unidade para a direita (KROESE e TAIMRE, 2018)

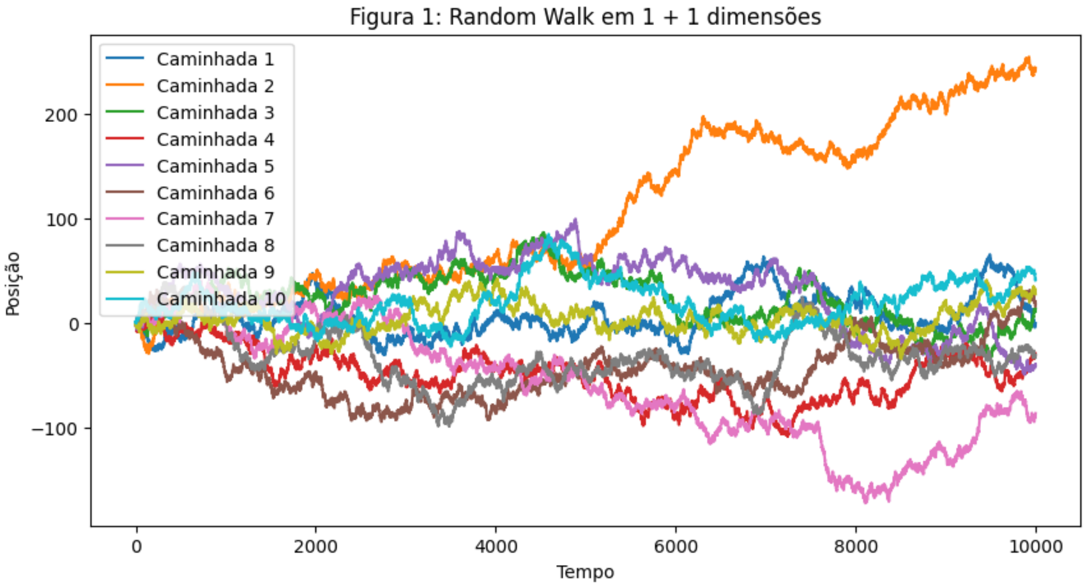
Ao repetir esse processo diversas vezes e calcular a posição final da partícula, podemos obter uma distribuição aproximada da posição final, calcular a média e o desvio padrão, entre outras estatísticas (VELTEN, 2009).

Uma medida utilizada em séries temporais é o coeficiente de Hurst, que avalia a dependência temporal. Ele é usado para estimar a auto-correlação em dados dependentes do tempo e pode ser usado para identificar se a série temporal exibe uma tendência de longo prazo (persistência) ou de curto prazo (anti-persistência) em seus movimentos (RESTA, 2012).

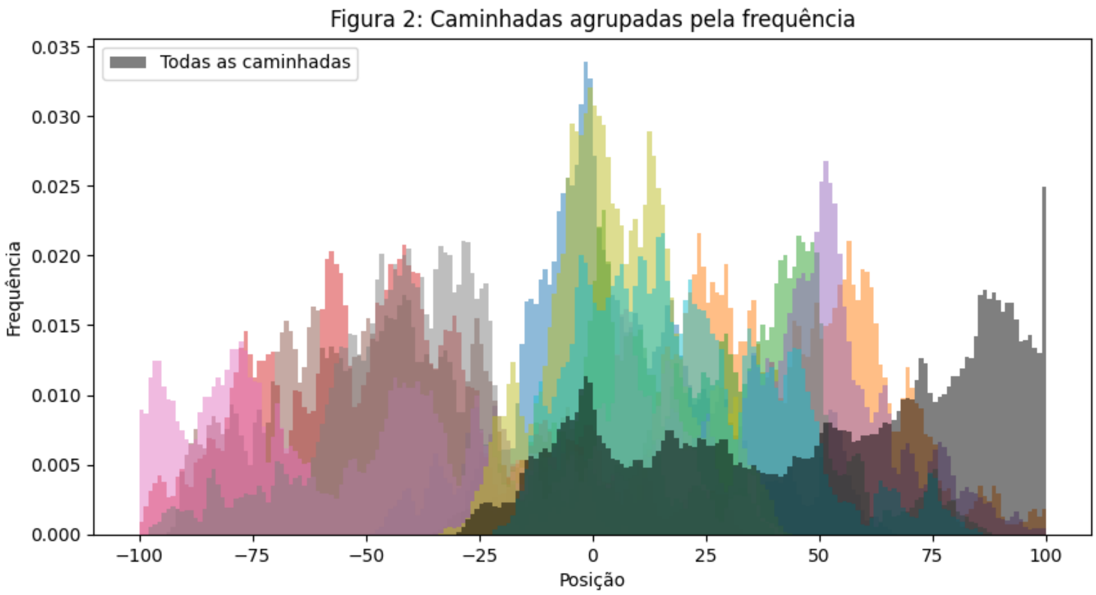
No contexto do modelo de Random Walk, o coeficiente de Hurst pode ser usado para verificar se os movimentos aleatórios gerados pelo modelo exibem dependência temporal ou não.

3- RESULTADOS

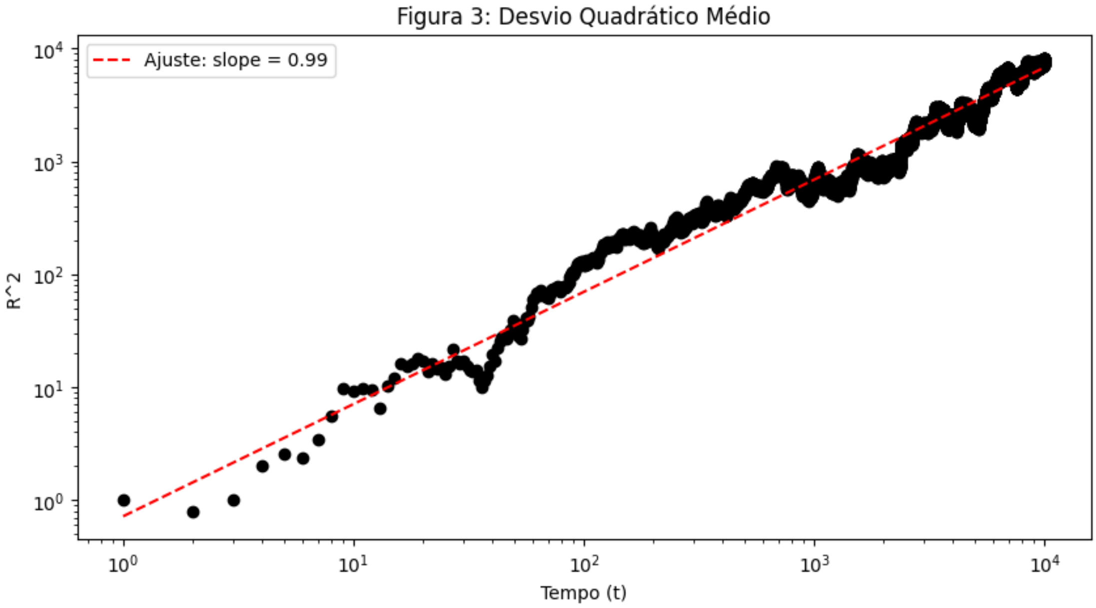
O algoritmo proposto foi executado 25 vezes, sendo 10 caminhadas com 10.000 passos cada uma. Assim, temos as 10 caminhadas conforme Figura 1, que mostra a relação *tempo vs posição*. O gerador congruencial é usada para gerar a sequência e definir os passos que serão dados em cada caminhada. Os passos são definidos como +1 se o número gerado for maior ou igual a 0.5, e -1 caso contrário.



A Figura 2 mostra as caminhadas agrupadas pela frequência em um gráfico de histograma. É possível verificar que as caminhadas geram sobreposições dos passos.



A Figura 3 mostra o desvio quadrático médio e o cálculo do coeficiente de ajuste da curva pela lei de potência. O gráfico tem escala $log - log$, em que o eixo *x* representa o tempo (t) e o eixo *y* representa o desvio quadrático médio (r^2). Também é calculado o ajuste de regressão linear no logaritmo dos dados ($np.log(t)$ e $np.log(r^2)$). Foi usando a função $np.polyfit$ com 1 grau, e os coeficientes da regressão linear (slope e intercept) são atribuídos às variáveis correspondentes. Em seguida, a curva ajustada é plotada ($'r--'$) no gráfico de escala log-log, usando a equação $np.exp(intercept) * t ** slope$, que representa a relação entre o tempo (t) e o desvio quadrático médio (r^2) de acordo com o ajuste de regressão linear.

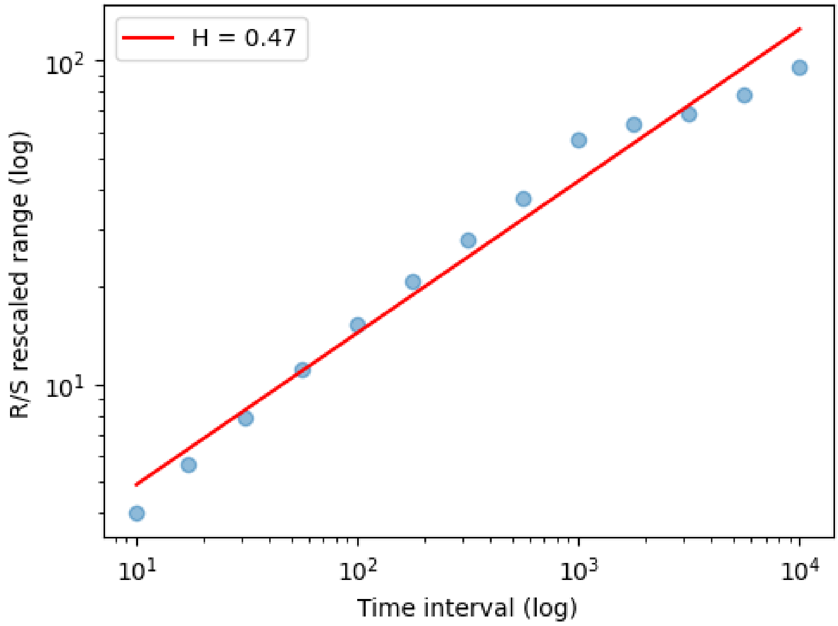


Foram calculadas as estatísticas como Curtose, Teste *t*, Expoente de Potência e *p-value*. A Curtose calculada foi de 4.06, a Estatística de Teste *t* foi calculado para o expoente da Lei de Potência com ajuste pela curtose foi de: -0.8748813411438668. A hipótese nula H_0 definida é de que o Random Walk é verdadeiro (slope = 1) com um nível de significância de 0.05. Nestes termos, aceita-se a hipótese nula. O *p-value* calculado foi de: 0.38165948830064345.

Posteriormente foi gerado a série temporal acumulada a partir das caminhadas aleatórias e calculado o coeficiente de Hurst de 0,47. A Figura 4 mostra o coeficiente de Hurst e a série temporal acumulada.

O código está disponível em:
https://github.com/mihanne/Praticas_PMMC/tree/main/Caminhante_Aleatorio

Figura 4: Série Temporal Acumulada



4- DISCUSSÃO

Os parâmetros definidos para a geração de números pseudoaleatórios foram definidos para melhor ajuste dos resultados. As constantes m, a, c e x0 são valores que influenciam na geração dos números aleatórios.

A variável m é um número grande que define o módulo da operação que será feita para gerar o próximo número aleatório. A variável a é um multiplicador utilizado para criar o próximo número aleatório a partir do número anterior. Já a variável c é um valor de incremento utilizado na geração do próximo número aleatório. Por fim, a variável x_0 é o valor inicial utilizado para iniciar a sequência de números aleatórios.

```
# Definindo as constantes e parâmetros
m = 2**32
a = 16807
c = 1013904223
x0 = 0

# Função para gerar números pseudo-aleatórios congruenciais
def congruential_random(n, m, a, c, x0):
    x = np.zeros(n)
    x[0] = x0
    for i in range(1, n):
        x[i] = (a*x[i-1] + c) % m
```

```
    return x/m
```

Na simulação do Random Walk primeiramente é criado um array t que representa o tempo, variando de 1 a n_steps. Em seguida, é calculado o desvio quadrático médio (r^2) das caminhadas aleatórias ao longo do tempo, usando a função $np.cumsum$ para calcular a soma cumulativa dos passos e depois elevando ao quadrado e tirando a média em relação aos diferentes caminhos aleatórios.

```
# Simulação do Random Walk em 1 + 1 dimensões
n_walks = 10
n_steps = 10000
steps = congruential_random(n_walks*n_steps, m, a, c, x0)
steps = np.where(steps < 0.5, -1, 1)
steps = np.reshape(steps, (n_walks, n_steps))
```

Em seguida, é calculado o ajuste de regressão linear no logaritmo dos dados ($np.log(t)$ e $np.log(r2)$) usando a função $np.polyfit$ com 1 grau, e os coeficientes da regressão linear. O ajuste da curva pela lei de potência foi de 0,99, ou seja, aproximadamente 1. Sendo assim, o resultado indica que a simulação indica um verdadeiro Random Walk.

```
# Cálculo do desvio quadrático médio e plot em escala log-log
t = np.arange(n_steps) + 1
r2 = np.mean(np.cumsum(steps, axis=1)**2, axis=0)
plt.figure(figsize=(10,5))
plt.title('Figura 3: Desvio Quadrático Médio')
plt.xlabel('Tempo (t)')
plt.ylabel('R^2')
plt.loglog(t, r2, 'ko')
slope, intercept = np.polyfit(np.log(t), np.log(r2), 1)
plt.loglog(t, np.exp(intercept)*t**slope, 'r--', la-
bel='Ajuste: slope = {:.2f}'.format(slope))
plt.legend()
```

As estatísticas calculadas durante a simulação são:

Curtose: 4.06

Aceita-se a hipótese nula H0 de que o Random Walk é verdadeiro (slope = 1) com um nível de significância de 0.05.

Expoente da lei de potência: 0.9935845177660209

Estatística de teste t: -0.8748813411438668

p-valor: 0.38165948830064345

A curtose é uma medida de quão achatada ou concentrada em torno da média é uma distribuição em relação a uma distribuição normal. Um valor de curtose maior do que 3 indica que a distribuição é mais concentrada em torno da média do que uma distribuição normal.

A estatística de teste t e o p-valor são usados para testar a hipótese nula de que o slope da linha ajustada no gráfico do desvio quadrático médio é igual a 1. O teste t é comparado com o valor crítico. Caso seja menor a hipótese nula é aceita, caso contrário é rejeitada.

```
if np.abs(t_statistic) < t_critical:
    print('Aceita-se a hipótese nula H0 de que o Random Walk é verda-
deiro (slope = 1) com um nível de significância de {:.2f}.'.format(alpha))
else:
    print('Rejeita-se a hipótese nula H0 de que o Random Walk é verda-
deiro (slope = 1) com um nível de significância de {:.2f}.'.format(alpha))
```

O expoente da lei de potência é uma medida da relação entre o desvio padrão e o tempo em uma simulação de Random Walk. Um valor de expoente próximo a 1 indica que o Random Walk é efetivamente aleatório, enquanto valores diferentes de 1 indicam diferentes graus de autocorrelação nos movimentos aleatórios.

Também foi calculado o coeficiente de Rust de 0,47, indicando anti-persistência. Em geral, se o coeficiente de Hurst estiver entre 0,5 e 1, isso indica que a série temporal é persistente, enquanto um valor entre 0 e 0,5 indica anti-persistência. Isso indica que o modelo pode ser uma boa representação do fenômeno em questão, uma vez que os movimentos aleatórios não estão correlacionados no tempo.

5- CONCLUSÃO

O algoritmo apresentado é uma implementação de um modelo de caminhada aleatória em uma dimensão, utilizando um gerador de números pseudo-aleatórios congruenciais.

São definidos os parâmetros do gerador (*m, a, c, x0)* e é criada a função "**congruential_random**" que utiliza esses parâmetros para gerar uma sequência de números pseudo-aleatórios.

Em seguida, o modelo de caminhada aleatória é simulado utilizando a sequência de números gerados. São geradas 10 caminhadas aleatórias em uma dimensão com 10000 passos cada uma.

As caminhadas são plotadas em um gráfico, onde é possível observar como cada caminhada se comporta em relação ao tempo. Em outro gráfico, as caminhadas são agrupadas de acordo com a frequência de cada posição e é gerado um histograma que mostra a distribuição de frequência.

Também é calculado o desvio quadrático médio das caminhadas e plotado em um gráfico em escala log-log. É ajustado um modelo de lei de potência para os dados, onde o expoente dessa lei representa a difusão da partícula na caminhada aleatória. É realizado um teste de hipótese para avaliar se o expoente da lei de potência é igual a 1, que seria o resultado esperado para uma caminhada aleatória verdadeira.

Além disso, o Teorema Central do Limite é verificado através do cálculo da distribuição das posições finais das caminhadas. É calculado também o momento de ordem 4 (curtose) das caminhadas e um teste de hipótese é realizado para avaliar se o expoente da lei de potência pode ser ajustado pela curtose.

Concluindo, o algoritmo implementa um modelo de caminhada aleatória em uma dimensão e utiliza diversas técnicas estatísticas para avaliar o comportamento do modelo que executa um Random Walk verdadeiro.

6- REFERÊNCIAS

DYM, Clive; HUPPENREUSS, Robert L. Principles of mathematical modeling. 2nd ed. San Diego: Academic Press, 2004.

KROESE, Dirk P.; TAIMRE, Thomas; BOTEV, Zdravko I. Introduction to Monte Carlo Methods. 2nd ed. New York: Springer, 2018.

RESTA, Marina. Hurst exponent and its applications in time-series analysis. Recent Patents on Computer Science, v. 5, n. 3, p. 211-219, 2012.

VELTEN, K. Mathematical Modeling and SImulation: Na Introduction for Scientists and Engineers Weinhein: Wiley-VCH, 2009.