SIMPÓSIO DAS EXATAS
UniFOA
2025

RELATO DE EXPERIÊNCIA

Encontrando o valor de pi com o Método de Monte Carlo em Python

Vitor Amadeu Souza[1]

1 – UniFOA, Centro Universitário de Volta Redonda, Volta Redonda, RJ.
vitor.amadeu@foa.org.br
0009-0002-1857-6799

Resumo: Este artigo apresenta uma implementação do método de Monte Carlo para aproximar o valor de π (pi) utilizando a linguagem de programação Python. O método baseia-se em princípios de probabilidade geométrica, onde pontos aleatórios são gerados dentro de um quadrado unitário, calculando-se a proporção daqueles que caem dentro de um círculo inscrito de raio unitário. Os resultados demonstram que o método de Monte Carlo constitui uma ferramenta educativa e intuitiva para aproximar π, evidenciando a relação entre simulações estatísticas e constantes matemáticas fundamentais. A implementação em Python permite visualizações do processo, facilitando a compreensão dos conceitos matemáticos subjacentes.

Palavras-chave: Aproximação de pi. Simulação Computacional. Probabilidade Geométrica. Simulação Estatística.

INTRODUÇÃO

O número π (pi) é uma das constantes matemáticas mais importantes e conhecidas, representando a razão entre o perímetro e o diâmetro de um círculo. Seu valor aproximado de 3,14159265359... tem intrigado matemáticos há milênios, com diversas metodologias desenvolvidas para sua aproximação (BAILEY; BORWEIN; CRANDALL, 2010).

Entre as abordagens computacionais para aproximar π, destaca-se o método de Monte Carlo, uma técnica de simulação estocástica que utiliza amostragem aleatória para obter resultados numéricos. Este método, desenvolvido durante o Projeto Manhattan na década de 1940 por cientistas como John von Neumann e Stanislaw Ulam, revolucionou a resolução de problemas matemáticos complexos (METROPOLIS, 1987).

A aproximação de π via Monte Carlo exemplifica perfeitamente a interseção entre probabilidade, geometria e computação. Como observado por Krauth (2006), este método constitui um caso didático ideal para compreender os fundamentos de simulações estocásticas devido à sua simplicidade conceitual e implementação direta.

A linguagem Python, com suas bibliotecas como NumPy e Matplotlib, proporciona um ambiente ideal para implementar e visualizar o método, tornando-o acessível tanto para fins educacionais quanto para pesquisa (VANDERPLAS, 2016). Segundo Guttag (2016), Python tem se destacado como ferramenta preferencial para computação científica devido à sua sintaxe clara e ecossistema robusto.

Este artigo apresenta uma análise completa da aplicação do método de Monte Carlo para aproximação de π, discutindo implementações em Python e convergência estatística.

METODOLOGIA

O princípio básico do método de Monte Carlo para aproximar π está fundamentado na relação entre as áreas de um círculo e um quadrado. Considerando um círculo de raio r inscrito em um quadrado de lado 2r, temos a relação apresentada na Tabela 1.

Tabela 1 - Relações entre o círculo e quadrado

Área do círculo	$Ac=\pi r^2$
Área do quadrado	$Aq= (2r)^2 = 4r^2$

Fonte: O Autor

A razão entre estas áreas é dada pela Tabela 2, onde é possível notar que π pode ser estimado como os pontos dentro do círculo pelo total de pontos.

Tabela 2 - Relações entre áreas do círculo e quadrado

Ac/Aq	$\pi r^2/4r^2=\pi/4$

Fonte: O Autor

Para a implementação do método de Monte Carlo em Python, utilizamos as bibliotecas NumPy para geração de números aleatórios e operações vetoriais, e Matplotlib para visualizações. O algoritmo segue os seguintes passos demonstrados na Tabela 3.

Tabela 3 - Passos para testar o algoritmo

Passo	Procedimento
1	Gerar n pontos aleatórios (x, y) dentro do quadrado unitário [0,1] × [0,1]
2	Contar quantos pontos caem dentro do círculo unitário centrado em (0.5, 0.5)
3	Calcular $\pi \approx 4 \times$ (pontos dentro do círculo / total de pontos)

Fonte: O Autor

O código-fonte para comparação de desempenho está disponível no link: https://github.com/vitor-souza-ime/pi_montecarlo, sendo que este código pode ser testado no Google Colab. Nesta versão, está implementado a versão para análise de convergência, registrando a aproximação de π conforme o número de pontos aumenta.

RESULTADOS E DISCUSSÃO

Foram realizadas simulações com diferentes quantidades de pontos, comparando as aproximações obtidas com o valor conhecido de π. Os resultados são apresentados na Tabela 4.

Tabela 4 - Aproximações de π para diferentes números de pontos.

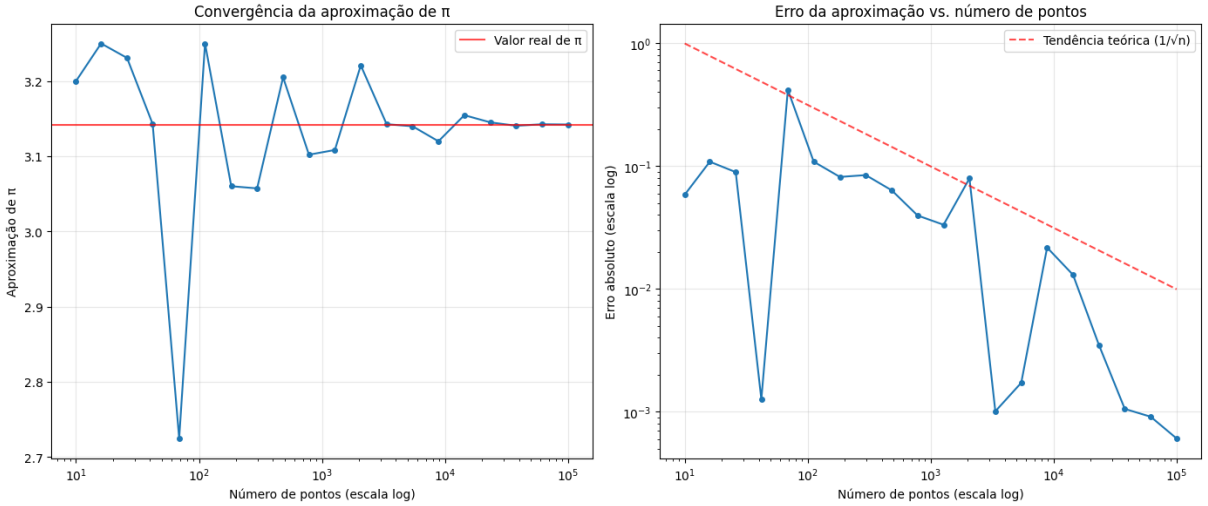
Número de pontos	Aproximação de π	Erro absoluto	Erro relativo
10^2	3.16	0.01841	0.586
10^3	3.152	0.01041	0.331
10^4	3.1368	0.00479	0.153
10^5	3.14208	0.00049	0.016
10^6	3.14182	0.00023	0.007
10^7	3.14154	0.00005	0.002

Fonte: O Autor

Os resultados demonstram que, conforme aumentamos o número de pontos, a aproximação converge para o valor real de π. Este comportamento confirma o teorema do limite central, que prevê que o erro diminua proporcionalmente à raiz quadrada do número de amostras (RIPLEY, 1987).

A Figura 1 mostra a convergência da aproximação em função do número de pontos, incluindo o intervalo de confiança de 95%. O gráfico de convergência revela que a variabilidade nas aproximações diminui significativamente após 10^5 pontos, corroborando a análise de erro apresentada na Tabela 4.

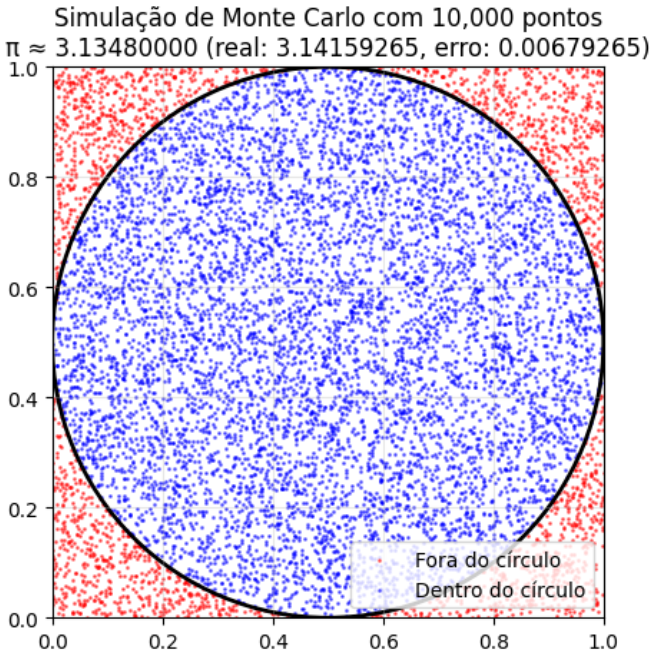
Figura 1 - Convergência da aproximação conforme aumenta n



Fonte: O autor

A Figura 2 apresenta um gráfico mostrando os pontos impressos dentro e fora do círculo, considerando o número de pontos de 10.000.

Figura 2 - Pontos dentro e fora do círculo para estimativa de π



Fonte: O autor

CONCLUSÕES

Este trabalho apresentou uma análise abrangente sobre a aplicação do método de Monte Carlo para aproximação do número π utilizando Python. O método de Monte Carlo constitui uma abordagem intuitiva e visualmente instrutiva para aproximar π, estabelecendo conexões significativas entre geometria, probabilidade e simulação numérica. Os resultados confirmam que a precisão da aproximação aumenta com a raiz quadrada do número de pontos simulados, demonstrando convergência estatística consistente com a teoria de probabilidade.

Embora existam métodos matematicamente mais eficientes para calcular π, o método de Monte Carlo permanece relevante como ferramenta educacional e como exemplo paradigmático de aplicação de técnicas de simulação estocástica. A visualização do processo de convergência facilita a compreensão intuitiva de conceitos estatísticos fundamentais, como a lei dos grandes números e o teorema do limite central, tornando este método particularmente valioso em contextos didáticos.

Para trabalhos futuros, sugere-se a exploração de implementações paralelas do algoritmo, possivelmente utilizando GPU via CUDA ou bibliotecas como Numba, permitindo simulações com bilhões de pontos. Também seria interessante comparar variantes do método de Monte Carlo que utilizam diferentes distribuições de amostragem, potencialmente melhorando a taxa de convergência.

REFERÊNCIAS

BAILEY, D. H.; BORWEIN, J. M.; CRANDALL, R. E. Advances in the theory of box integrals. Mathematics of Computation, v. 79, n. 271, p. 1839–1866, 2010. DOI: https://doi.org/10.1016/j.jsc.2013.09.001.

METROPOLIS, N. The beginning of the Monte Carlo method. Los Alamos Science, n. 15, p. 125-130, 1987.

RIPLEY, B. D. Stochastic Simulation. New York: John Wiley & Sons, 1987.

VANDERPLAS, J. Python Data Science Handbook: Essential Tools for Working with

Data. Sebastopol: O'Reilly Media, 2016.

GUTTAG, J. V. Introduction to Computation and Programming Using Python: With Application to Understanding Data. Cambridge: MIT Press, 2016.

KRAUTH, W. Statistical Mechanics: Algorithms and Computations. Oxford: Oxford University Press, 2006.