

# Estudo sobre a relação entre desvio padrão da média e tempo de execução em simulações de Monte Carlo

Mayara Yumi Ikeda

## 1. Introdução

O método de Monte Carlo consiste em usar dados com natureza aleatória para obter resultados determinísticos. Um quarto de círculo foi inscrito em um quadrado de raio  $r = 1$  (Figura 1). O programa define pontos  $(x, y)$  de forma aleatória no intervalo  $[0, 1]$ . O número  $\pi$  pode ser escrito em termos da razão  $\rho$  entre a área do círculo  $\frac{\pi r^2}{4}$  e a área do quadrado  $r^2$ .

$$\rho = \frac{\pi r^2}{4r^2} = \frac{\pi}{4} \rightarrow \pi = 4\rho \quad (1)$$

O método tenta aproximar a razão entre as áreas pela razão do número de sucessos  $n$ , que é definido como a quantidade de pontos  $(x, y)$  que caem embaixo da curva referente ao círculo, e o número total de pontos usados na simulação  $N$ . Para  $N$  grande, da ordem de centenas, a aproximação é razoável.

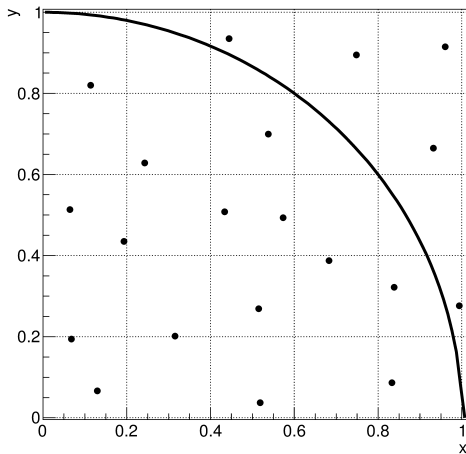


Figura 1: Uma tentativa de ilustrar o algoritmo. A função densidade de probabilidade que descreve o número de sucessos  $n$  é uma binomial.

Foram simulados  $N_* = 1000$  valores, com diferentes  $N$ , variando de 100 a 1100000 pontos. Se espera que um  $N$  maior implique numa menor incerteza para a média dos valores simulados. A incerteza nos dá uma intuição do quanto o valor simulado está próximo do valor verdadeiro. Pela natureza do método, a função densidade de probabilidade que descreve os valores de  $\pi$  simulados pelo método é uma binomial.

A complexidade de tempo do algoritmo usado, é  $O(n)$ , ou seja, o tempo cresce aproximadamente de forma linear com o número de pontos simulados.

```
1 for (int j=0; j<no_simulacoes; j++){
2     dentro=0;
3
4     for (int i=0; i<no_pontos; i++){
5         x= dis(prng); y= dis(prng);
6
7         if (y<=sqrt(1-pow(x,2))) {
8             dentro++;
9         }
10    }
11    pi = 4*double(dentro)/double(no_pontos);
```

Listing 1: Simulação Monte Carlo em C++ usado para simular 1000 valores para  $\pi$ .

Será mostrado que a incerteza da média dos  $N_*$  valores simulados é uma função bem conhecida de  $N$ :

$$\sigma_{\langle \pi \rangle}(N) = \frac{4}{\sqrt{N_*}} \frac{\sqrt{0.17N}}{N}$$

## 2. Fundamentos teóricos

O método de Monte Carlo aproxima a razão entre as áreas do círculo e quadrado pela razão entre o número de sucessos  $n$  pelo número total de pontos simulados  $N$ :

$$\rho = \frac{n}{N} \quad 3$$

Substituindo (3) em (1):

$$\pi_{\text{simulado}} = 4 \frac{n}{N} \quad 4$$

A F.D.P que descreve a probabilidade do número de sucessos  $n$  para uma simulação de  $N$  tentativas é uma binomial. Assim sendo, a incerteza de  $n$ ,  $\sigma_n$  é dada por:

$$\sigma_n = \sqrt{Np(1-p)} \quad 5$$

Onde  $p \sim 0.785$  é a probabilidade de sucesso,  $\frac{\pi}{4}$ . O valor de  $p(1-p)$  é aproximadamente 0.17.

Por propagação de incertezas, a incerteza de cada  $\pi_{simulado}$  é dada por:

$$\sigma_{\pi_{simulado}} = \frac{4}{N} \sqrt{Np(1-p)} \quad 6$$

A simulação é repetida  $N_*$  vezes, a incerteza do valor médio de  $\pi_{simulado}$ ,  $\sigma_{\langle \pi \rangle}$  é dada pelo desvio padrão da média:

$$\begin{aligned} \sigma_{\langle \pi \rangle} &= \frac{\sigma_{\pi_{simulado}}}{\sqrt{N_*}} \\ &= \frac{4}{\sqrt{N_*}} \frac{\sqrt{0.17N}}{N} \end{aligned} \quad 7$$

Escrevendo N como sendo  $N(t) = \alpha t$  e substituindo em (7):

$$\sigma_{\langle \pi \rangle}(t) = 4 \sqrt{\frac{0.17}{N_* \alpha t}} \quad 8$$

### 3. Metodologia

O experimento foi realizado no mesmo computador, pois é preferível que o experimento seja reprodutível. Informações relevantes sobre as simulações feitas estão contidas na tabela abaixo:

Tabela 1: Simulações estudadas

Simulações Monte Carlo, $N_* = 1000$				
N	$\langle \pi \rangle$	$\sigma_{\langle \pi \rangle}$	Tempo t (s)	$\sigma_t$
$10^2$	3.136	0.005	0.1058	0.0001
$10^3$	3.142	0.001	0.9966	0.0003
$10^4$	3.1412	0.0005	9.814	0.001
$10^5$	3.1413	0.0002	98.26	0.09
$2 \times 10^5$	3.1417	0.0001	201.5	2.0
$3 \times 10^5$	3.14151	0.00009	296.4	2.9
$4 \times 10^5$	3.14152	0.00008	394.9	3.1
$5 \times 10^5$	3.14155	0.00007	493.7	4.9
$6 \times 10^5$	3.14162	0.00006	596.5	6.0
$7 \times 10^5$	3.14163	0.00006	691.3	6.9
$8 \times 10^5$	3.14153	0.00005	789.1	7.9
$9 \times 10^5$	3.14161	0.00005	888.2	8.9
$10^6$	3.14170	0.00005	979.1	0.4
$1.1 \times 10^6$	3.14148	0.00004	1086.8	10.9

#### 3.1. Medindo o tempo de execução do programa

Uma das maiores dificuldades iniciais foi encontrar uma forma de medir o tempo e conseguir uma incerteza para esse medida. O comando *time* do *bash* do Linux possui uma precisão limitada, na casa dos milisegundos. Além disso era necessário digitar o comando várias vezes para conseguir alguma incerteza, o que é tedioso. A solução encontrada para esse problema

foi usar uma ferramenta usada chamada *perf*, que possibilita a medida dos tempos das simulações de forma mais precisa (na ordem dos nanossegundos), e avaliar as incertezas mais facilmente, pois essa ferramenta devolve uma incerteza associada ao tempo médio depois de  $c$  vezes em que o programa é executado. Vale comentar que dados foram analisados preliminarmente e foi constatado que as incertezas dos tempos estavam subestimadas. Foram feitas correções nas incertezas referentes ao tempo de execução, multiplicando-as por  $\sqrt{\chi_{red}^2}$ .

#### 3.2. Encontrando a relação entre N e t

A hipótese é a de N é uma função linear de t, pela complexidade de tempo do algoritmo. Um gráfico de N vs. t foi *plotado*, os resultados estão na Figura 2:

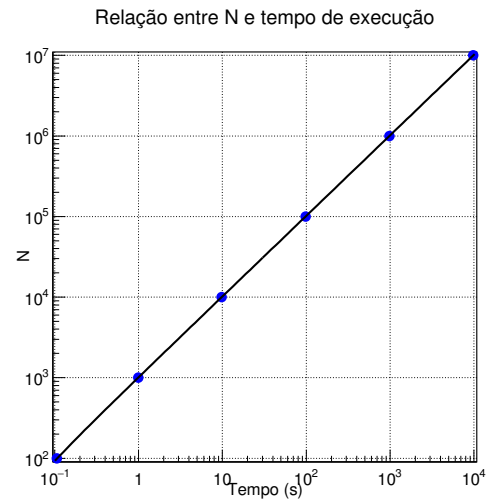


Figura 2: O gráfico mostra a relação linear entre N e t. Note a escala log em x e y.

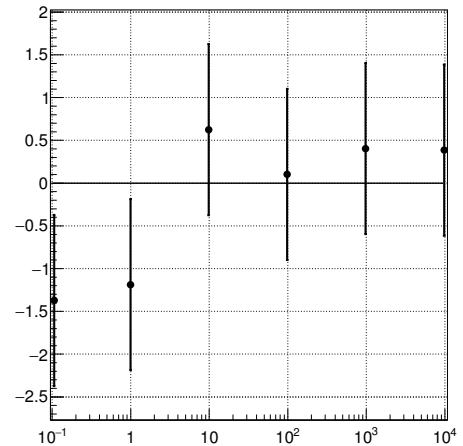


Figura 3: Gráfico de resíduos referente ao gráfico mostrado na Figura 2

O  $\chi^2$  é bom quando comparado ao número de graus de liberdade. Assim N pode ser escrito levando em consideração os

parâmetros ajustados:

Tabela 2: Parâmetros ajustados no gráfico mostrado na Figura 2

Parâmetros ajustados	
$\alpha$	$1014.9 \pm 4.8$
$\chi^2/\text{NDF}$	4/4

$$N(t) = (1014.9 \pm 4.8)t$$

### 3.3. Ajustando a função teórica aos dados

A expressão (8) mostra como as incertezas das médias tomadas por meio das simulações se comportam com relação ao tempo. Para fins de comparação a função mostrada em (9) foi ajustada aos dados, onde [0] é o parâmetro que se deseja encontrar por meio do ajuste:

$$\sigma_{\langle \pi \rangle}(t) = \frac{[0]}{\sqrt{t}} \quad 9$$

$$[0] \rightarrow 4 \sqrt{\frac{0.17}{N_* \alpha}} \sim 0.0016 \pm 0.0039$$

As incertezas das médias foram calculadas analiticamente por meio de (7), e foi feita uma comparação entre os parâmetros ajustados nos  $\sigma_{\langle \pi \rangle_{teorico}}$  e  $\sigma_{\langle \pi \rangle_{exp}}$ .

## 4. Resultados

A expressão (9) foi ajustada nos conjuntos de dados. A seguir estão os gráficos juntamente com as tabelas:

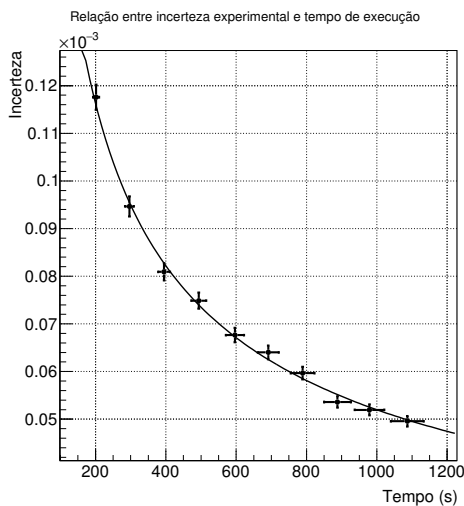


Figura 4: Ajuste da expressão (10) sobre os desvios padrões das médias simuladas para  $\pi$

A derivada da função ajustada no gráfico mostrado na Figura 6 nos revela que a variação da incerteza com relação ao tempo

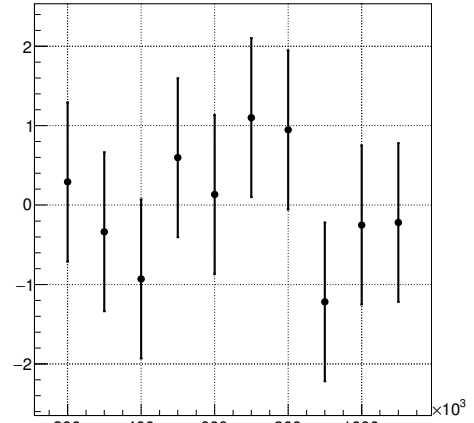


Figura 5: Gráfico de resíduos referente ao gráfico mostrado na Figura 4

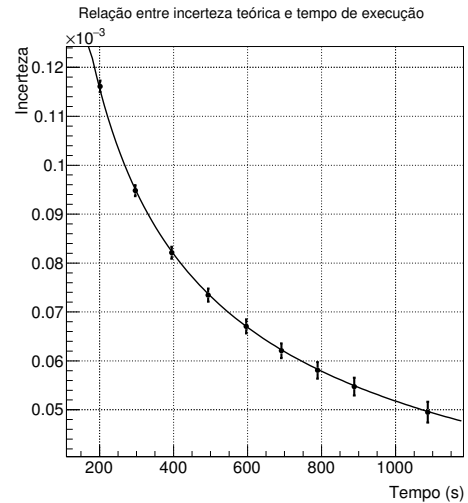


Figura 6: Ajuste da expressão (10) sobre os desvios esperados para  $\pi$ . Vide expressão (7)

Tabela 3: Parâmetros ajustados no gráfico mostrado na Figura 4

Parâmetros ajustados	
[0]	$0.00163 \pm 0.00002$
$\chi^2/\text{NDF}$	2.97 / 9

Tabela 4: Parâmetros ajustados no gráfico mostrado na Figura 6

Parâmetros ajustados	
[0]	$0.001634 \pm 0.000009$
$\chi^2/\text{NDF}$	0.62 / 8

tende a zero, quando o tempo tende a infinito (Figura 8), para conseguir incertezas cada vez menores é necessário intervalos cada vez maiores de tempo. Portanto o método de Monte Carlo não possui aplicações práticas no cálculo de  $\pi$ , outros métodos (como usar a série de Leibniz) convergem mais rapidamente para o valor de  $\pi$ . Por outro lado, Monte Carlo nos dá uma

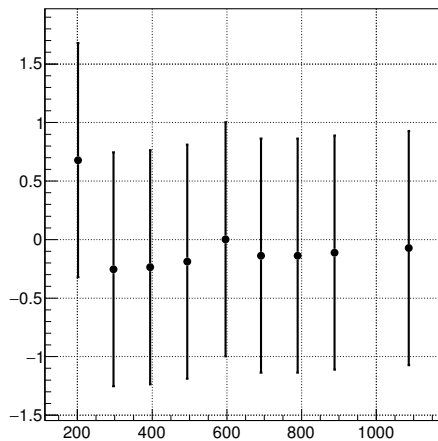


Figura 7: Gráfico de resíduos referente ao gráfico mostrado na Figura 6

tempo de execução do programa todo, o que inclui por exemplo o output dos valores simulados para um arquivos de dados, e o modelo relaciona apenas o tempo da simulação com as incertezas. Por mais que o tempo de execução de  $n$  instruções seja determinístico, uma série de fatores fora do alcance estão influenciando os resultados. Seria quase como soltar um objeto, acionar o cronômetro e medir o intervalo de tempo decorrido *dezenas de minutos* depois do objeto ter chegado ao chão (e essas dezenas de minutos não são sempre as mesmas!).

incerteza para cada valor simulado, ou seja, nos dá uma ideia da qualidade da aproximação, ao contrário dos métodos que envolvem soma de termos.

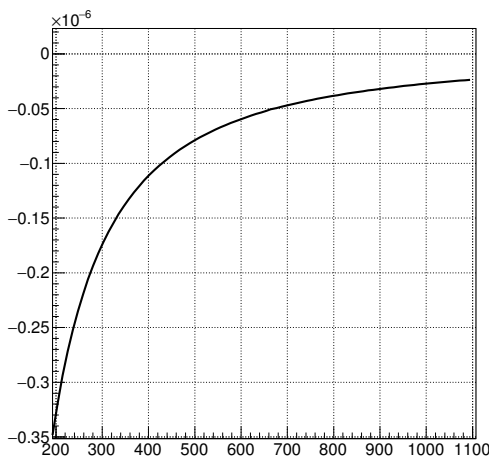


Figura 8: Derivada do ajuste referente à Figura 6.

## 5. Discussão dos resultados

Os parâmetros ajustados aos dados experimentais são compatíveis com o valor esperado, levando em consideração as incertezas para o tempo, com  $z_{experimental} = 0.70$  e  $z_{teorico} = 0.04$ .

Os ajustes dos gráficos mostrados nas Figuras 2 e 6 não descreve bem o conjunto de dados, (vide gráfico de resíduos) os pontos tendem a estar abaixo do gráfico, indicando que o modelo precisa ser revisto. Isso é esperado, sendo que os gráficos estão vinculados. Por outro lado o ajuste ao gráfico mostrado na Figura 6 é bastante satisfatório. O método realizado para determinar o tempo de simulação possui falhas: o tempo medido é o