



UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA

EM703 - Instrumentação

Projeto Final

Aproximação Mecânica para π

Lucas Menezes Barreto - 220846
Paulo Yoshio Kuga - 204451
Rafael Augusto Braga Cota - 243366
Thiago Rodrigues Timóteo - 244776
Vinícius Yochida Ramacciotti - 188294

Campinas
Julho 2022

Sumário

1 Revisão Probabilística	2
1.1 Função Geradora de Momentos	2
1.2 Distribuição Binomial	2
1.2.1 Aproximação de Stirling	3
1.2.2 Aproximação Normal para a Binomial	6
2 Discussão Matemática do Problema	8
2.1 Caso $l < d$	8
2.1.1 Cômputo Probabilístico	8
2.1.2 Número de Lançamentos Necessários	9
2.2 Caso $l > d$	10
2.2.1 Cômputo Probabilístico	10
2.2.2 Número de Lançamentos Necessários	10
3 Simulação Computacional	13
3.1 Números pseudoaleatórios	13
3.1.1 O entendimento da aleatoriedade	13
3.1.2 Geradores Congruentes Lineares (GCL)	13
3.2 Algoritmo Mersenne Twister	14
3.2.1 Breve introdução	14
3.2.2 Analogia para explicação do algoritmo	14
3.3 Discussão da implementação	15
3.3.1 Escolha da linguagem	15
3.4 Discussão da implementação	16
3.5 Resultados da computação	16
4 Procedimento Experimental	17
4.1 Descrição do experimento e análises	17
4.2 Discussão do procedimento experimental - análises e incertezas	21
5 Conclusões Gerais	22
6 Anexo A: Códigos	23
6.1 Cômputo de $N(a)$	23
6.2 Código da simulação	23
6.3 Código de análise dos resultados da simulação	25
7 Referências	26

1 Revisão Probabilística

1.1 Função Geradora de Momentos

Para traduzir um resultado, é conveniente definir uma variável aleatória X como uma função que a cada evento físico associa um elemento do contradomínio em \mathbb{R} . Se um problema de interesse está especificado por duas coordenadas, seja $C \subset \mathbb{R}^2$ o contradomínio de um possível evento E_1 , ou seja $X(E_1) \times Y(E_1) = C$, onde X, Y duas variáveis aleatórias e o produto é cartesiano. X, Y são ditas conjuntamente contínuas se existir uma função Darboux-integrável tal que a probabilidade de que ocorra C seja dada por:

$$P(E_1) = P((X, Y) \in C) = \int \int_C f(x, y) dx dy \quad (1.1)$$

Ademais, dizemos que X, Y são independentes se houver duas funções tais que $f(x, y) = f_X(x)f_Y(y)$. Voltamos para o caso unidimensional. Se houver uma variável aleatória X , enxergamos a esperança de que X ocorra como o valor do contradomínio ponderado pela probabilidade (que no caso contínuo é $f(x)dx$).

$$E[X] = \int_{-\infty}^{\infty} xf(x) dx \quad (1.2)$$

No caso discreto, temos uma definição essencialmente igual, exceto pelo fato de que o contradomínio de X admite alguns valores apenas e neste caso:

$$E[X] = \sum_i x_i p(X = x_i) \quad (1.3)$$

Definimos uma nova variável aleatória por $Y = e^{tX}$ e escrevemos a função geradora como:

$$\Phi_X(t) = E[Y] = E[e^{tX}] = \sum_i e^{tx_i} p(X = x_i) \quad (1.4)$$

Derivando-se em relação a t e calculando em 0 a derivada, obtém-se facilmente que: $\Phi'_X(0) = E[X]$ e $\Phi''_X(0) = E[X^2]$. Disso decorre que:

$$\begin{aligned} E[(X - E[X])^2] &= \sum_i (x_i^2 - 2x_i E[X] + E[X]^2)p(X = x_i) = E[X^2] - 2E[X]^2 + E[X]^2 \Rightarrow \\ E[(X - E[X])^2] &= \Phi''_X(0) - \Phi'_X(0) \end{aligned} \quad (1.5)$$

Esta última esperança é calculada em uma variável aleatória indicativa da dispersão em relação à média e em geral denota-se $\text{var}[X] = E[(X - E[X])^2]$ referida por variância. Também é costumeiro denotar a raiz da variância por $\sigma = \sqrt{\text{var}[X]}$, conhecida por desvio padrão.

1.2 Distribuição Binomial

Seja uma sucessão de n eventos e $X = k$ quando houver k sucessos dentre as n tentativas. Aqui a palavra sucesso indica simplesmente que tenha ocorrido algum evento de interesse. Neste caso, a distribuição é dita normal se $p(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$, onde p é a probabilidade de sucesso quando $n = 1$. Pela expressão (1.4), obtemos que:

$$\Phi_X(t) = \sum_k e^{tk} p(X = k) = \sum_k e^{tk} \binom{n}{k} p^k (1-p)^{n-k} = \sum_k \binom{n}{k} (pe^t)^k (1-p)^{n-k} = (pe^t + 1 - p)^n \quad (1.6)$$

Dos nossos conhecimentos sobre a função geradora de momentos, obtemos:

$$\Phi'_X(0) = E[X] = \frac{\partial}{\partial t}(pe^t + 1 - p)^n \Big|_{t=0} = n(pe^t + 1 - p)^{n-1}pe^t \Big|_{t=0} = n1^{n-1}p = np \quad (1.7)$$

Além disso a esperança de $E[X^2]$ pode ser calculada usando a derivada segunda de Φ_X , ou seja:

$$\begin{aligned} \Phi''_X(0) &= E[X^2] = \frac{\partial^2}{\partial t^2}(pe^t + 1 - p)^n \Big|_{t=0} = \frac{\partial}{\partial t}n(pe^t + 1 - p)^{n-1}pe^t \Big|_{t=0} = \\ &(n(n-1)(pe^t + 1 - p)^{n-2}p^2e^{2t} + n(pe^t + 1 - p)^{n-1}pe^t) \Big|_{t=0} = n(n-1)p^2 + np = n^2p^2 + np - np^2 \end{aligned}$$

Por fim, invocando a equação 1.5, podemos computar $\text{var}[X]$, que fica:

$$\text{var}[X] = n^2p^2 + np - np^2 - (np)^2 = np(1-p) \quad (1.8)$$

1.2.1 Aproximação de Stirling

Esta seção terá por objetivo demonstrar que $n! \sim \sqrt{2\pi n}n^n e^{-n}$, ou, mais precisamente, que $\lim_{n \rightarrow \infty} \frac{n!}{\sqrt{2\pi n}n^n e^{-n}} = 1$. O leitor que aceitar este fato como verdade, poderá continuar para próxima seção sem perda de continuidade.

Lema 1.1.

$$\int \ln(x)dx = n \ln(n) - n + 1$$

Prova: De fato, isso decorre de calcular explicitamente a integral:

$$\begin{aligned} \int \ln(x)1dx &= x \ln(x) - \int \frac{1}{x}xdx = x \ln(x) - x \\ \int_1^n \ln(x)dx &= n \ln(n) - n - (1 \ln 1 - 1) = n \ln(n) - n + 1 \end{aligned}$$

□

Lema 1.2. Para $x > 0$, tem-se a seguinte desigualdade

$$\ln(1+x) > \frac{2x}{2+x}$$

Prova: Defina a função $g(x) = \ln(1+x) - \frac{2x}{2+x}$. De calcular a derivada obtém-se:

$$g'(x) = \frac{x^2}{(1+x)(2+x)^2}$$

$g'(x)$ é claramente maior do que 0 para $x > -1$, de onde obtemos que a função é crescente. Em particular, se $x > 0$, $g(x) > g(0) = 0$.

□

Lema 1.3. A sequência $a_n = \frac{\sqrt{n}n^n e^{-n}}{n!}$ é convergente.

Prova: De fato, basta mostrar que ela é crescente e limitada superiormente, porque neste caso, ela convergirá para o supremo de acordo com nossos conhecimentos de análise. Tome $x = \frac{1}{n}$ na desigualdade do lema 1.2. Neste caso:

$$\begin{aligned} \ln\left(1 + \frac{1}{n}\right) &> \frac{2\frac{1}{n}}{2 + \frac{1}{n}} = \frac{2}{2n+1} \Rightarrow \left(n + \frac{1}{2}\right) \ln\left(1 + \frac{1}{n}\right) > 1 \Rightarrow \\ \ln\left(1 + \frac{1}{n}\right)^{n+\frac{1}{2}} &> 1 \Rightarrow \left(1 + \frac{1}{n}\right)^{n+\frac{1}{2}} = \left(1 + \frac{1}{n}\right)^n \sqrt{1 + \frac{1}{n}} > e \Rightarrow \\ \frac{1}{e} \left(1 + \frac{1}{n}\right)^n \sqrt{1 + \frac{1}{n}} &> 1 \end{aligned}$$

Por outro lado, de fazer a razão a_{n+1}/a_n , fica claro que:

$$\frac{a_{n+1}}{a_n} = \frac{\sqrt{n+1}(n+1)^{n+1}e^{-(n+1)}}{(n+1)!} \frac{n!}{\sqrt{n}n^n e^{-n}} = \sqrt{\frac{n+1}{n}} \frac{(n+1)^n}{n^n e} = \frac{1}{e} \left(1 + \frac{1}{n}\right)^n \sqrt{1 + \frac{1}{n}} > 1$$

Isso estabelece que a_n é de fato uma função crescente. Para mostrar que se trata de uma função limitada, note que pela imagem abaixo a área entre $\ln(x)$ entre inteiros consecutivos fica limitada superiormente pela área do trapézio.

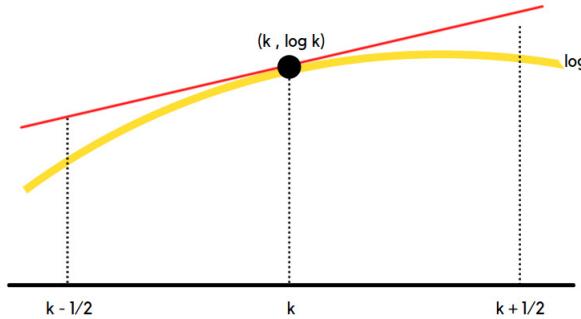


Figura 1: Limitante Superior da Área

A reta tangente é dada por $y(x) = \ln(k) + \frac{1}{k}(x - k)$. Pela convexidade, $\ln(x) \leq y(x)$, em particular no intervalo $[k - \frac{1}{2}, k + \frac{1}{2}]$. Assim:

$$\begin{aligned} \int_{k-\frac{1}{2}}^{k+\frac{1}{2}} \ln(x) dx &\leq \int_{k-\frac{1}{2}}^{k+\frac{1}{2}} \ln(k) + \frac{x-k}{k} dx = \ln(k) \Rightarrow \\ \int_{\frac{3}{2}}^{n-\frac{1}{2}} \ln(x) dx &= \sum_{k=2}^{n-1} \int_{k-\frac{1}{2}}^{k+\frac{1}{2}} \ln(x) dx \leq \sum_{k=2}^n \ln(k) \end{aligned}$$

Claramente vale também que $\int_1^{\frac{3}{2}} \ln(x) dx \leq (\frac{3}{2} - 1)2 = 1$ e também $\int_{n-\frac{1}{2}}^n \ln(x) dx \leq \frac{1}{2} \ln(n)$. Portanto:

$$\int_1^n \ln(x) dx \leq 1 + \sum_{k=2}^{n-1} \ln(k) + \frac{1}{2} \ln(n)$$

Pelo lema 1.1, obtém-se que a integral pode ser reescrita de modo a obter:

$$\begin{aligned}
n \ln(n) - n + 1 &\leq 1 + \sum_{k=2}^{n-1} \ln(k) + \frac{1}{2} \ln(n) \Rightarrow \\
n \ln(n) - n - \ln(n!) - \frac{1}{2} \ln(n) + \ln(n) &\leq 0 \\
\ln(a_n) = \frac{1}{2} \ln(n) + n \ln(n) - n - \ln(n!) &\leq 0 \Rightarrow a_n \leq 1
\end{aligned}$$

Assim, a sequência também é limitada e isso encerra a demonstração. \square

Lema 1.4. (*Produto de Wallis*)

$$\prod_{n=1}^{\infty} \frac{2n}{2n-1} \frac{2n}{2n+1} = \frac{\pi}{2}$$

Prova: Em primeiro lugar defina a sequência cujo n -ésimo termo é dado por $S_n = \int_0^{\frac{\pi}{2}} (\sin(x))^n dx$. Calculando diretamente, obtém-se que $S_1 = 1$ e $S_2 = \frac{\pi}{4}$. Provaremos por indução que:

$$\begin{aligned}
S_{2k} &= \frac{1 \cdot 3 \cdot \dots \cdot (2k-1)}{2 \cdot 4 \cdot \dots \cdot (2k)} \frac{\pi}{2} \\
S_{2k+1} &= \frac{2 \cdot 4 \cdot \dots \cdot (2k)}{1 \cdot 3 \cdot \dots \cdot (2k+1)}
\end{aligned}$$

Os casos base estão demonstrados. Além disso, se $n > 2$, de integração por partes fica evidente que:

$$\begin{aligned}
S_n &= \int_0^{\frac{\pi}{2}} -\sin^{n-1}(x) \frac{d}{dx} \cos(x) dx = -\cos(x) \sin^{n-1}(x) \Big|_0^{\frac{\pi}{2}} + (n-1) \int_0^{\frac{\pi}{2}} \cos^2(x) \sin^{n-2}(x) dx = \\
&= (n-1) \int_0^{\frac{\pi}{2}} \sin^{n-2}(x) dx - (n-1) \int_0^{\frac{\pi}{2}} \sin^n(x) dx = (n-1)S_{n-1} + (1-n)S_n \Rightarrow \\
S_n &= \frac{n-1}{n} S_{n-2}
\end{aligned}$$

Suponha que valha para S_{2k} , então pela fórmula recursiva acima:

$$S_{2k+2} = \frac{2k+1}{2k+2} \frac{1 \cdot 3 \cdot \dots \cdot (2k-1)}{2 \cdot 4 \cdot \dots \cdot (2k)} \frac{\pi}{2}$$

Similarmente, suponha que valha para S_{2k+1} , então pela fórmula recursiva demonstrada:

$$S_{2k+3} = \frac{2k+2}{2k+3} \frac{2 \cdot 4 \cdot \dots \cdot (2k)}{1 \cdot 3 \cdot \dots \cdot (2k+1)}$$

Assim, por indução as fórmulas precisam estar coerentes. Note adicionalmente que no intervalo de integração $0 \leq \sin(x) \leq 1 \Rightarrow 0 \leq \sin^{n+1}(x) \leq \sin^n(x) \Rightarrow 0 \leq S_{n+1} \leq S_n, \forall n$. Assim, S_n é decrescente e limitada inferiormente, de onde se conclui que ela seja convergente para o ínfimo, de acordo com os nossos conhecimentos de análise. Além disso:

$$\frac{2k}{2k+1} = \frac{S_{2k+1}}{S_{2k-1}} \leq \frac{S_{2k+1}}{S_{2k}} \leq 1 \Rightarrow \lim_{k \rightarrow \infty} \frac{S_{2k+1}}{S_{2k}} = 1$$

$$\prod_{n=1}^{\infty} \frac{2n}{2n-1} \frac{2n}{2n+1} = \lim_{k \rightarrow \infty} \frac{\pi}{2} \frac{S_{2k+1}}{S_{2k}} = \frac{\pi}{2} 1 = \frac{\pi}{2}$$

\square

Teorema 1.1. (*Aproximação Assintótica*)

$$n! \sim \sqrt{2\pi n} n^n e^{-n}$$

Prova:

$$\prod_{n=1}^k \frac{2n}{2n-1} \frac{2n}{2n+1} = \frac{2 \cdot 2 \cdot 4 \cdot 4 \cdot 6 \cdot 6 \cdot \dots \cdot (2k) \cdot (2k)}{3 \cdot 3 \cdot 5 \cdot 5 \cdot \dots \cdot (2k-1) \cdot (2k-1) \cdot (2k+1)} =$$

$$\frac{2^4 \cdot 4^4 \cdot 6^4 \cdot \dots \cdot (2k)^4}{2^2 \cdot 3^2 \cdot 4^2 \cdot 5^2 \cdot \dots \cdot (2k-1)^2 (2k)^2 (2k+1)} = \frac{(2 \cdot 4 \cdot 6 \cdot \dots \cdot (2k))^4}{(2 \cdot 3 \cdot 4 \cdot 5 \cdot \dots \cdot (2k-1) \cdot (2k))^2 (2k+1)} = \frac{2^{4k} (k!)^4}{(2k!)^2 (2k+1)}$$

Desse algebrismo em associação com o lema 1.4, obtemos:

$$\lim_{k \rightarrow \infty} \sqrt{\prod_{n=1}^k \frac{2n}{2n-1} \frac{2n}{2n+1}} = \lim_{k \rightarrow \infty} \frac{2^{2k} (k!)^2}{2k! \sqrt{2k+1}} = \sqrt{\frac{\pi}{2}}$$

Por outro lado, fica claro que $\sqrt{2k+1} = \sqrt{k} \sqrt{2 + \frac{1}{k}}$. Assim:

$$\lim_{k \rightarrow \infty} \frac{2^{2k} (k!)^2}{2k! \sqrt{k}} = \lim_{k \rightarrow \infty} \frac{2^{2k} (k!)^2}{2k! \sqrt{k}} \frac{1}{\sqrt{2 + \frac{1}{k}}} \lim_{k \rightarrow \infty} \sqrt{2 + \frac{1}{k}} = \sqrt{\frac{\pi}{2}} \sqrt{2} = \sqrt{\pi}$$

Na mesma notação do lema 1.3 sabemos que a_n converge e como $a_n \geq \frac{1}{e} = a_1 > 0 \Rightarrow \lim_{n \rightarrow \infty} a_n > 0$. Adicionalmente:

$$\frac{a_n^2}{a_{2n}} = \frac{\left(\frac{\sqrt{n} n^n e^{-n}}{n!}\right)^2}{\left(\frac{\sqrt{2n} (2n)^{2n} e^{-2n}}{2n!}\right)} = \frac{n \cdot n^{2n} e^{-2n}}{(n!)^2} \frac{(2n)!}{\sqrt{2n} (2n)^{2n} e^{-2n}} = \frac{1}{\sqrt{2}} \frac{\sqrt{n} (2n)!}{2^{2n} (n!)^2} \Rightarrow$$

$$\lim_{n \rightarrow \infty} \frac{a_n^2}{a_{2n}} = \frac{1}{\sqrt{2}} \lim_{n \rightarrow \infty} \left(\frac{2^{2n} (n!)^2}{2n! \sqrt{n}}\right)^{-1} = \frac{1}{\sqrt{2\pi}} \Rightarrow$$

$$\lim_{n \rightarrow \infty} \frac{\sqrt{n} n^n e^{-n}}{n!} = \lim_{n \rightarrow \infty} \frac{a_n^2}{a_{2n}} \frac{a_n}{a_{2n}} = \frac{1}{\sqrt{2\pi}} \frac{L}{L} = (\sqrt{2\pi})^{-1} \Rightarrow$$

$$\lim_{n \rightarrow \infty} \frac{\sqrt{2\pi} \sqrt{n} n^n e^{-n}}{n!} = 1 \Rightarrow n! \sim \sqrt{2\pi n} n^n e^{-n}$$

□

1.2.2 Aproximação Normal para a Binomial

Mostraremos que para n suficientemente grande, a distribuição binomial é bem aproximada pela distribuição contínua Gaussiana (ou Normal). Isso será interessante no estudo da agulha de Buffon para estimar o número de lançamentos necessários para obter uma estimativa razoável para π .

Teorema 1.2. (*Aproximação Normal*)

$$\binom{n}{k} p^k (1-p)^{n-k} \sim \frac{1}{\sqrt{2\pi \text{var}[X]}} e^{\frac{-(k - E[X])^2}{2\text{var}[X]}}$$

Prova: Pelo teorema 1.1, ganhamos a aproximação assintótica de Stirling, de modo que:

$$\begin{aligned} \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k} &\approx \frac{n^n e^{-n} \sqrt{2\pi np^k (1-p)^{n-k}}}{k^k e^{-k} \sqrt{2\pi k} (n-k)^{n-k} e^{-(n-k)} \sqrt{2\pi(n-k)}} = \\ n^n \left(\frac{p}{k}\right)^k \left(\frac{1-p}{n-k}\right)^{n-k} \sqrt{\frac{n}{k2\pi(n-k)}} &= \left(\frac{np}{k}\right)^k \left(\frac{n(1-p)}{n-k}\right)^{n-k} \sqrt{\frac{n}{2\pi k(n-k)}} = \\ \sqrt{\frac{n}{2\pi k(n-k)}} e^{k \ln(\frac{np}{k}) + (n-k) \ln(\frac{n(1-p)}{n-k})} &= \sqrt{\frac{n}{2\pi k(n-k)}} e^{-k \ln(\frac{k}{np}) + (k-n) \ln(\frac{(n-k)}{n(1-p)})} \end{aligned}$$

Por outro lado, sabemos que k não pode fugir muito de np , uma vez que np é a esperança pela equação 1.7. De fato, para k suficientemente longe de np , digamos t desvios padrões, tanto a binomial como a gaussiana convergem para zero e não há nada a mostrar. Além disso, $k - np$ tem ordem de no máximo \sqrt{n} , porque o desvio padrão é da forma $\sqrt{np(1-p)}$ de modo que no máximo $k = np \pm t\sqrt{np(1-p)}$. Feita esta digressão, note que por expansão em série de Taylor em torno de 1:

$$\begin{aligned} \ln\left(\frac{k}{np}\right) &= \ln\left(1 + \frac{k-np}{np}\right) \approx \frac{k-np}{np} - \frac{1}{2} \left(\frac{k-np}{np}\right)^2 \\ \ln\left(\frac{n-k}{n(1-p)}\right) &= \ln\left(1 + \frac{np-k}{n(1-p)}\right) \approx \frac{np-k}{n(1-p)} - \frac{1}{2} \left(\frac{np-k}{n(1-p)}\right)^2 = \end{aligned}$$

Assim, o expoente pode ser aproximado por:

$$\begin{aligned} -k \left(\frac{k-np}{np} - \frac{1}{2} \left(\frac{k-np}{np}\right)^2\right) + (k-n) \left(\frac{np-k}{n(1-p)} - \frac{1}{2} \left(\frac{np-k}{n(1-p)}\right)^2\right) \\ (np-k-np) \left(\frac{k-np}{np} - \frac{1}{2} \left(\frac{k-np}{np}\right)^2\right) + (k-np+n(p-1)) \left(\frac{np-k}{n(1-p)} - \frac{1}{2} \left(\frac{np-k}{n(1-p)}\right)^2\right) = \\ -\frac{(k-np)^2}{np} - \frac{(np-k)^2}{n(1-p)} - (k-np) - (np-k) + \frac{1}{2} \frac{(k-np)^3}{(np)^2} + \frac{1}{2} \frac{(k-np)^2}{np} + \\ \frac{1}{2} \frac{(np-k)^3}{n^2(1-p)^2} + \frac{1}{2} \frac{(np-k)^2}{n(1-p)} = -\frac{(k-np)^2}{2npq} + \frac{1}{2} \frac{(np-k)^3}{n^2(1-p)^2} + \frac{1}{2} \frac{(k-np)^3}{(np)^2} \end{aligned}$$

Pelo que discutimos $(np - k)^3$ cresce na pior das hipóteses como $\alpha n^{\frac{3}{2}}$. O denominador cresce mais rapidamente. Assim, a exponencial fica bem aproximada por:

$$e^{-k \ln(\frac{k}{np}) + (k-n) \ln(\frac{(n-k)}{n(1-p)})} \sim e^{-\frac{(k-np)^2}{2npq}} \quad (1.9)$$

Trabalhemos a raiz para fazê-la mais próxima do que queremos:

$$\begin{aligned} \sqrt{\frac{n}{2\pi k(n-k)}} &= \sqrt{\frac{n}{2\pi(k-np+np)(n(1-p)+np-k)}} = \\ \sqrt{\frac{1}{2\pi(np(1-p)+(k-np)(1-2p)-\frac{(k-np)^2}{n})}} &= \sqrt{\frac{1}{2\pi n(p(1-p)+\frac{(k-np)}{n}(1-2p)-\frac{(k-np)^2}{n^2})}} \end{aligned}$$

Os termos de $(k-np)^a$ crescem como $\sqrt{n^a}$. Ambos os denominadores suplantam esse crescimento. Assim o que prepondera é o termo livre que $k - np$, de modo que podemos escrever:

$$\sqrt{\frac{n}{2\pi k(n-k)}} \sim \sqrt{\frac{1}{2\pi np(1-p)}} \quad (1.10)$$

Combinando as duas aproximações assintóticas em 1.9 e 1.10, bem como as equações 1.8 e 1.7, obtemos a condição requerida. \square

2 Discussão Matemática do Problema

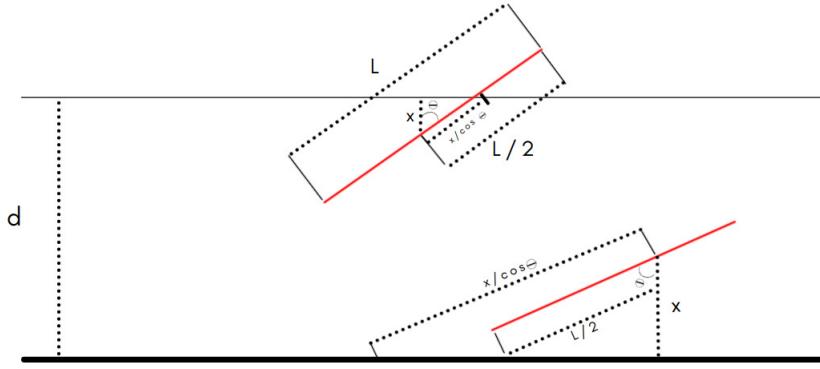


Figura 2: Variáveis de Interesse

Considere uma haste de tamanho l que esteja posicionada em um plano graduado por retas que distam d entre si. Defina X a variável aleatória que a cada posição da haste associa a distância x do centro de massa até a linha mais próxima. Analogamente, defina Y a variável aleatória que a cada posição da agulha associa o valor θ que se dá pelo menor ângulo determinado pela haste e pela reta que passa pelo centro de massa e é perpendicular às graduações do plano.

Essas duas medidas supomos independentes (o valor de θ e de x podem variar livres um do outro.) Além disso, note que pela simetria $x \in [0, \frac{d}{2}]$ e também, $\theta \in [0, \frac{\pi}{2}]$. Se não houver uma preferência de orientação, podemos associar uma densidade de probabilidade constante para cada uma das variáveis aleatória. Ou seja:

$$f(x, \theta) = f_X(x)f_\Theta(\theta) = \frac{1}{d/2} \frac{1}{\pi/2} = \frac{4}{d\pi} \quad (2.1)$$

Nota-se pela figura que a hipotenusa do triângulo formado pela agulha possui valor $\frac{x}{\cos \theta}$, desde que $\theta \neq \frac{\pi}{2}$. Nesta condição de θ , uma intersecção entre a agulha e uma das graduações do plano ocorre se e somente se $\frac{l}{2} > \frac{x}{\cos(\theta)}$. Se $\theta = \frac{\pi}{2}$, então a intersecção ocorre somente se $x = 0$. Entretanto, a probabilidade em um único par $(0, \frac{\pi}{2})$ possui valor nulo para distribuições não concentradas. Portanto da equação 1.1, a probabilidade de que ocorra um cruzamento com a linha é:

$$p = \frac{4}{\pi d} \int \int_{\frac{l}{2} > \frac{x}{\cos(\theta)}} dx d\theta \quad (2.2)$$

Por outro lado, sob uma perspectiva frequentista, tem-se que depois de N lançamentos com n cruzamentos da haste a probabilidade fica bem aproximada pela razão, ou seja:

$$\frac{n}{N} \approx \frac{4}{\pi d} \int \int_{\frac{l}{2} > \frac{x}{\cos(\theta)}} dx d\theta \Rightarrow \pi = \frac{4N}{nd} \int \int_{\frac{l}{2} > \frac{x}{\cos(\theta)}} dx d\theta \quad (2.3)$$

2.1 Caso $l < d$

2.1.1 Cômputo Probabilístico

A desigualdade requerida é equivalente a pedir $\frac{l}{2} \cos(\theta) > x$. A figura abaixo mostra os pontos (x, θ) que satisfazem a desigualdade requerida dentro do retângulo do domínio $[0, \frac{d}{2}] \times [0, \frac{\pi}{2}]$.

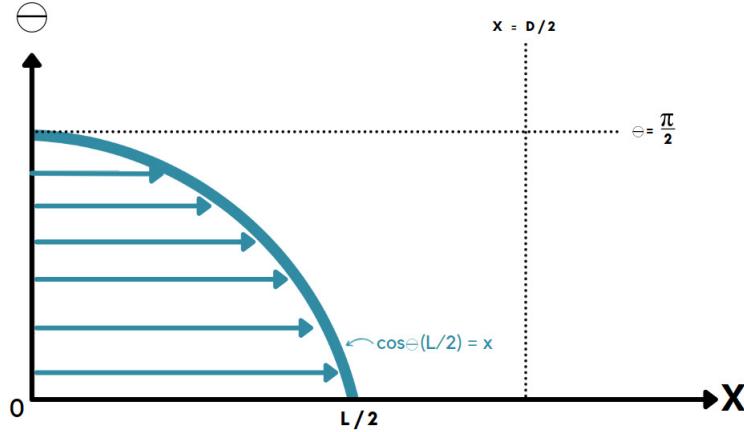


Figura 3: Área de Integração em $l < d$

Dessa forma, a equação 2.2 fica calculada como:

$$p = \frac{4}{\pi d} \int_0^{\frac{\pi}{2}} \int_0^{\frac{l}{2} \cos(\theta)} dx d\theta = \frac{2l}{\pi d} \int_0^{\frac{\pi}{2}} \cos(\theta) d\theta = \frac{2l}{\pi d} \quad (2.4)$$

2.1.2 Número de Lançamentos Necessários

Note que temos uma situação com probabilidade p de sucesso com N lançamentos repetidos e independentes com n sucessos. Logo, trata-se da distribuição binomial. Além disso, pelas equações 1.7 e 2.4, a esperança é que haja $Np = \frac{2Nl}{\pi d}$ sucessos. Se quisermos ter 95% de chance de que o erro do n/N ser menor do que ϵ escrevemos:

$$p \left(\left| n - \frac{2Nl}{\pi d} \right| < N\epsilon \right) > 0,95$$

Entretanto pelo teorema 1.2 (aproximação normal), obtemos que:

$$p(n = z) \approx \int_{z-\frac{1}{2}}^{z+\frac{1}{2}} \frac{1}{\sqrt{2\pi \text{var}[X]}} e^{-\frac{(k-E[X])^2}{2\text{var}[X]}} dx$$

Assim, a probabilidade que queremos calcular coincide com a probabilidade de que estejamos na gaussiana entre $\pm \frac{N\epsilon}{\sigma} = \pm \frac{N\epsilon}{\sqrt{Np(1-p)}}$ desvios padrões da média. Portanto:

$$\frac{N\epsilon}{\sqrt{Np(1-p)}} = 1,96 \Rightarrow N^{-1} = \frac{\epsilon^2}{1,96^2 p(1-p)} = \frac{\epsilon^2 \pi^2 d^2}{1,96^2 \cdot 2l(\pi d - 2l)} \quad (2.5)$$

Esta última equação é reveladora porque pode-se enxergar a quantidade razoável do número de lançamentos como dependente da constante adimensional $a = \frac{d}{l}$, precisamente $N(a) = \frac{1,96^2 \cdot 2(\pi a - 2)}{\epsilon^2 \pi^2 a^2}$. Se quisermos que a função atinja um mínimo ou máximo de N , podemos estudar as derivadas. Chamando $\xi = \frac{\epsilon^2 \pi^2}{2 \cdot 1,96^2}$

$$N'(a) = \frac{-2\xi a(\pi a - 2) + \pi \zeta a^2}{(\epsilon^2 \pi^2 a^2)^2} = 0 \Rightarrow \xi a^2 \pi = 4\xi a \Rightarrow a = \frac{4}{\pi}$$

Pela expressão da derivada, nota-se que a derivada é positiva para valores menores do que a e negativa para valores maiores. Disso decorre que este é um ponto de máximo local e a maneira menos otimizada de conduzir o experimento é que $d \approx 1,273l$. Tomando-se $\epsilon = 0,01$, obtemos que uma quantidade necessária de lançamentos é de $N = 4222$ para o nosso caso particular em que $\frac{d}{l} = 5,08$. Atribuindo-se que a incerteza $\delta n = N\epsilon$, obtemos de calcular a propagação na equação 2.3:

$$\delta\pi = \sqrt{\left(\frac{2N\delta l}{dn}\right)^2 + \left(\frac{2Nl\delta d}{d^2n}\right)^2 + \left(\frac{2NlN\epsilon}{dn^2}\right)^2} \quad (2.6)$$

2.2 Caso $l > d$

2.2.1 Cômputo Probabilístico

Neste caso, ainda temos uma área de integração semelhante à interior, exceto pelo fato de que nem todo x sob a curva pode ser aproveitado. Note que, de fato $x \in [0, \frac{d}{2}]$.

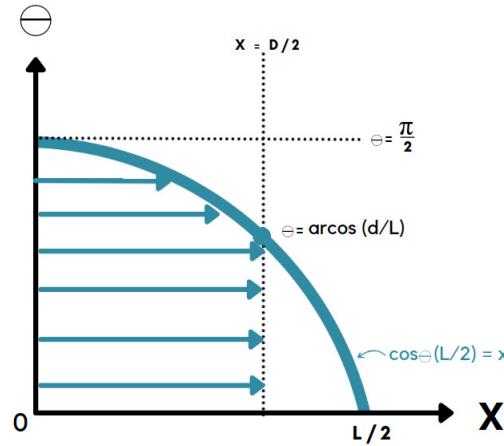


Figura 4: Área de Integração em $l > d$

Pela figura, podemos separar a região de integração em duas partes, uma das quais é um retângulo de área $\frac{d}{2} \arccos\left(\frac{d}{l}\right)$:

$$\begin{aligned} p &= \frac{4}{\pi d} \left(\frac{d}{2} \arccos\left(\frac{d}{l}\right) + \int_{\arccos\left(\frac{d}{l}\right)}^{\frac{\pi}{2}} \int_0^{\frac{l \cos(\theta)}{2}} dx d\theta \right) = \\ &\frac{2}{\pi} \arccos\left(\frac{d}{l}\right) + \frac{4}{\pi d} \frac{l}{2} \left(\sin\left(\frac{\pi}{2}\right) - \sin\left(\arccos\left(\frac{d}{l}\right)\right) \right) = \frac{2}{\pi} \arccos\left(\frac{d}{l}\right) + \frac{2l}{\pi d} \left(1 - \sqrt{1 - \frac{d^2}{l^2}} \right) \Rightarrow \\ p &= \frac{2}{\pi} \arccos\left(\frac{d}{l}\right) + \frac{2l}{\pi d} \left(1 - \sqrt{1 - \frac{d^2}{l^2}} \right) \end{aligned} \quad (2.7)$$

2.2.2 Número de Lançamentos Necessários

Como no caso anterior, podemos calcular a esperança de n sucessos de uma distribuição binomial. Desta vez obtém-se uma expressão um pouco mais complexa:

$$Np = \frac{2N}{\pi} \arccos\left(\frac{d}{l}\right) + \frac{2Nl}{\pi d} \left(1 - \sqrt{1 - \frac{d^2}{l^2}}\right)$$

Novamente, caso queiramos que a probabilidade de n/N fugir ao valor exato (a menos de ϵ) seja menor do que 5% impomos que pela distribuição Gaussiana haja:

$$p\left(\left|n - \frac{2N}{\pi} \arccos\left(\frac{d}{l}\right) - \frac{2Nl}{\pi d} \left(1 - \sqrt{1 - \frac{d^2}{l^2}}\right)\right| < N\epsilon\right) > 0,95$$

Essa condição será atendida se e somente se não se fugir mais do que $N\epsilon/\sigma$ desvios padrões da média. Portanto, como no caso anterior, precisamos que N seja pelo menos:

$$\begin{aligned} N^{-1} &= \frac{\epsilon^2}{1,96^2 p(1-p)} = \\ \frac{\epsilon^2}{1,96^2} &\frac{1}{\frac{2}{\pi} \arccos\left(\frac{d}{l}\right) + \frac{2l}{\pi d} \left(1 - \sqrt{1 - \frac{d^2}{l^2}}\right)} \frac{1}{1 - \frac{2}{\pi} \arccos\left(\frac{d}{l}\right) + \frac{2l}{\pi d} \left(1 - \sqrt{1 - \frac{d^2}{l^2}}\right)} \end{aligned} \quad (2.8)$$

Para enxergar o que ocorre em ambos os casos, ao variar o parâmetro $a = \frac{d}{l}$, foi feito o seguinte gráfico:

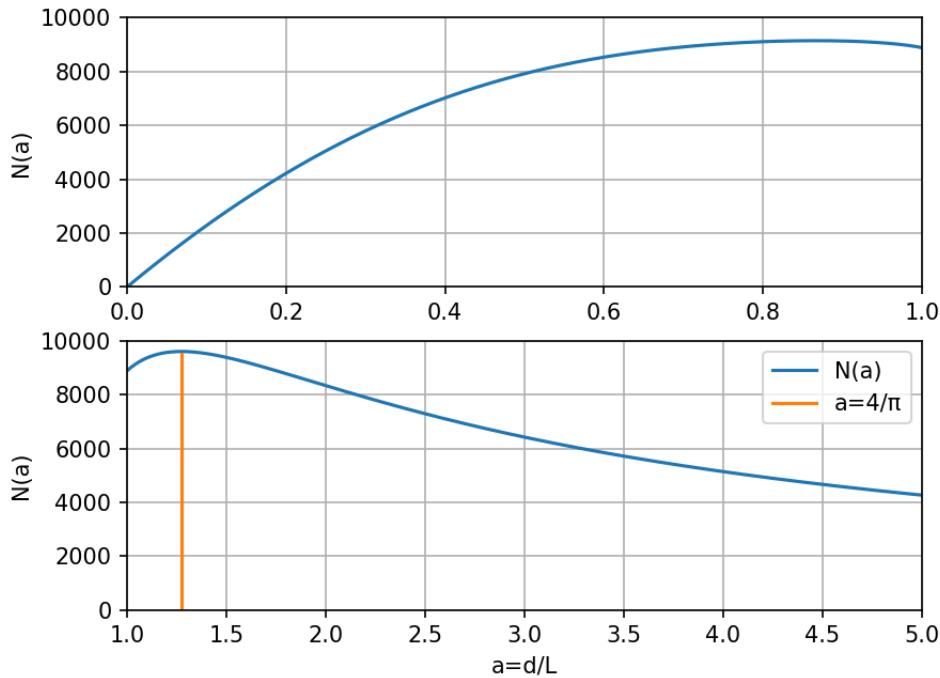


Figura 5: Representação gráfica das equações 2.5 e 2.8.

De rearranjar a equação 2.7, obtemos a equação requerida para o cômputo de π . Propagaremos a incerteza nela:

$$\pi = \frac{2N}{n} \arccos\left(\frac{d}{l}\right) + \frac{2Nl}{nd} \left(1 - \sqrt{1 - \frac{d^2}{l^2}}\right) \Rightarrow$$

$$\begin{aligned}
(\delta\pi)^2 = & \left(\frac{2N}{n^2} \arccos\left(\frac{d}{l}\right) N\epsilon + \frac{2Nl}{n^2 d} \left(1 - \sqrt{1 - \frac{d^2}{l^2}}\right) N\epsilon \right)^2 + \\
& \left(\frac{2N}{n} \frac{-1}{\sqrt{1 - (d/l)^2}} \frac{1}{l} - \frac{2Nl}{nd^2} \left(1 - \sqrt{1 - \frac{d^2}{l^2}}\right) + \frac{2Nl}{nd} \left(1 - \frac{1}{2} \left(1 - \frac{d^2}{l^2}\right)^{-1/2} \frac{-2d}{l^2}\right) \right)^2 \delta d^2 + \\
& \left(\frac{2N}{n} \frac{-1}{\sqrt{1 - (d/l)^2}} \frac{-d}{l^2} + \frac{2N}{nd} \left(1 - \sqrt{1 - \frac{d^2}{l^2}}\right) + \frac{2Nl}{nd} \left(1 - \frac{1}{2} \left(1 - \frac{d^2}{l^2}\right)^{-1/2} \frac{2d^2}{l^3}\right) \right)^2 \delta l^2 \quad (2.9)
\end{aligned}$$

3 Simulação Computacional

3.1 Números pseudoaleatórios

3.1.1 O entendimento da aleatoriedade

Em uma simulação probabilística é necessário que possamos gerar lançamentos aleatórios. Para isto temos de gerar números que representem os estados do problema, de tal forma que possamos estabelecer as devidas computações para chegarmos à convergência de um valor.

Devemos entender a aleatoriedade como sendo definida pela qualidade que estabelece que um fenômeno é imprevisível. Porém, a aleatoriedade vem através de uma perspectiva. Como estipula a questão do experimento mental do Demônio de Laplace, se um ser hipotético conhecer todos os estados do universo e o seu modelo, ele poderá ser capaz de prever o próximo estado. Desta forma, neste caso, para este ser, nada seria aleatório, uma vez que ele pode conhecer todos os próximos estados apenas com a informação do estado presente.

Dentro de um computador, é possível que nós conheçamos todos os estados em virtude da própria estrutura da memória do computador e a definição matemática de qualquer algoritmo que seja possível executar no mesmo (Máquina de Turing). Por isso, em computadores digitais, temos uma dificuldade de estabelecer números realmente aleatórios, uma vez que, o computador digital só estabelece dois estados para um determinado endereço de memória. Desta forma, como todos os estados são determinados, não há forma com a qual possamos realmente gerar ou obter de dentro da máquina, um sinal indeterminado.

Um exemplo de problema é que um computador digital é, de certa forma, um “Demônio de Laplace” de uma máquina virtual que ele mesmo roda. Esta implicação é necessária para que o computador possa emular outra máquina. Se esta máquina gerasse sinais aleatórios em sua perspectiva, para o computador, não seria aleatório, uma vez que este conheceria a razão que qualificaria aquele sinal.

3.1.2 Geradores Congruentes Lineares (GCL)

Para isto introduz-se a ideia do pseudorrandômico, que tem a capacidade de estabelecer uma distribuição de números similar a uma distribuição de números obtidos a partir de uma causa aleatória. Esta distribuição é gerada a partir de estados determinados, por isso não se pode chamá-la de realmente randômica.

Para tal, o método mais simples que se encontra na literatura são os Geradores Congruentes Lineares, que podemos ver como uma regra de geração de números a partir de um estado inicial.

$$X_{n+1} \equiv (aX_n + c) \pmod{p} \quad (3.1)$$

Lembremos que congruência modular significa somente dizer que $p|(X_{n+1} - (aX_n + c))$, ou seja, p é capaz de dividir a subtração dos elementos congruentes. A partir deste valor, estabelece-se uma iteração da fórmula e obtemos vários valores pseudoaleatórios que seguem uma distribuição uniforme. De certa forma, como estamos aplicando em computadores, para provar a distribuição, é bem vindo um teste empírico.

Em (7), KNUTH discute várias implementações que foram testadas a partir deste modelo básico de GCL. A existência de 4 parâmetros que podem ser alterados, gera uma grande dificuldade na hora de escolher e ”afiná”-los, visto que se, por exemplo, tomamos c como sendo 0, temos um subconjunto especial que auxilia na hora da computação em si (a nível de hardware), porém, acaba convergindo no número inicial (seed) muito rapidamente. Outra implementação citada é a utilização de um X_n e um X_{n+j} associados, que em outras palavras, seriam tomar dois números aleatórios da sequência e gerar outro número aleatório. Porém, essa aleatoriedade de j é arbitrária, ou seja, quem escolhe é o programador. Desta forma, são muitas possibilidades a serem escolhidas.

Como queremos escolher um gerador de caráter prático, nossa estratégia de busca foi entender qual gerador é implementado em linguagens modernas de programação. Quando utilizamos a função

`rand()` da biblioteca “random” da linguagem Python por exemplo, muitas vezes o usuário apenas aceita o resultado. Ele não tem ciência, muito menos tempo (ou interesse) de saber como esta função foi implementada. Porém, no nosso trabalho, como estamos lidando com o entendimento do aleatório, é mister entender como vamos obter nosso número randômico. Desta forma, estabelece-se uma avergiuação na documentação do Python, onde procuramos qual algoritmo gera o resultado da função `rand()`. Analisando a documentação da biblioteca “random”, ela refere-se a utilização do Algoritmo Mersenne Twister.

3.2 Algoritmo Mersenne Twister

3.2.1 Breve introdução

O Algoritmo Mersenne Twister é um algoritmo lançado em 1997 por Makoto Matusmoto e Takuji Nishimura no paper “Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator”. Diferente dos Geradores Congruentes Lineares, o Mersenne Twister é capaz, em apenas uma única iteração, gerar 624 números aleatórios a partir de um estado inicial de 32 bits. Podemos enxergar também este grande número aleatório como um número de 19968 bits, (porém isso não tem nenhuma utilização prática).

3.2.2 Analogia para explicação do algoritmo

O artigo propõe uma explicação do algoritmo que em essência tem uma abordagem matemática, no sentido que explora uma visão se utilizando da Álgebra Linear em um conjunto $\mathbb{F}_2 = \{0, 1\}$. No entanto, por estarmos no contexto da Instrumentação, mais convém explicar pelo ferramental que nós temos de variáveis de Estado nos reais. Desta forma, lendo o artigo é possível avaliar que os autores explicitamente se utilizam do conceito de estados (praticamente o mesmo que temos no livro de Katsuhiko Ogata [8]) para avaliar o algoritmo. Assim, propomos nos explicar por uma analogia.

Vamos supor uma equação diferencial de ordem w em \mathbb{R} em função de uma variável t .

$$\sum_{i=0}^w a_i \cdot y^{(i)}(t) = 0 \quad (3.2)$$

Note, que de acordo com [8], podemos tomar o modelo de estados como:

$$\vec{X} = \begin{cases} x_0 = y \\ x_1 = y^{(1)} \\ x_2 = y^{(2)} \\ \vdots \\ x_{n-1} = y^{(n-1)} \end{cases} \quad (3.3)$$

Desta forma, na representação de estados temos:

$$\dot{\vec{X}} = A \cdot \vec{X} \quad (3.4)$$

E temos A como sendo:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{w-1} & a_{w-2} & a_{w-3} & a_{w-4} & \dots & a_0 \end{bmatrix} \quad (3.5)$$

Porém, para uma computação numérica desta equação, é preciso tornar esta equação diferencial no domínio discreto. Vamos chamar este domínio de \mathbb{D} . Portanto, a equação 3.4 se torna:

$$\begin{cases} \dot{\vec{X}} = \frac{\vec{X}_{k+1} - \vec{X}_k}{\Delta t} \\ \dot{\vec{X}} = A \cdot \vec{X}_k \end{cases} \quad (3.6)$$

$$\frac{\vec{X}_{k+1} - \vec{X}_k}{\Delta t} = A \cdot \vec{X}_k \quad (3.7)$$

$$\vec{X}_{k+1} = \vec{X}_k + (A \cdot \vec{X}_k) \cdot \Delta t \quad (3.8)$$

Esta formulação é bem conhecida, principalmente em Controle de Sistemas. Nossa analogia parte do fato de que podemos, como dito na seção 3.1.1, analisar a memória do computador como estados, teríamos de certa forma o fato de que, para um número qualquer que pode ser representado como um vetor de \mathbb{F}_2 , sua existência é diretamente correlacionada a de seu anterior, de tal forma que ela realmente depende do número de uma ordem menor que ela.

Posteriormente, analisamos que se estamos trabalhando em \mathbb{F}_2 , o único valor aceitável para Δt é 1. Não obstante, estamos somando neste conjunto. Portanto precisamos tirar o módulo. Os autores convencionam o símbolo da soma com módulo 2 como sendo \oplus Logo temos:

$$\vec{X}_{k+1} = \vec{X}_k \oplus A \cdot \vec{X}_k \quad (3.9)$$

Dentro de \mathbb{F}_2 esta operação é simplificada, de tal forma que podemos fazer apenas operações simples com bits (por exemplo “shiftright”), obtendo o vetor \vec{X}_{k+1} de uma forma muito rápida. Porém, para adaptar a condições de uma distribuição uniforme dentro do valor de bits da estrutura de dados primitiva (estas condições são melhor detalhadas em KNUTH [7]), precisamos multiplicar pela matriz de têmpora T pela direita, tendo portanto $\vec{Z} = \vec{X} \cdot T$. Em T temos o que os autores chamam de “máscaras” para os bits, que tornam mais difíceis de serem perceptíveis o estado anterior. Desta forma, Z é o nosso resultado final. No caso, a matriz T, valores de a, e as máscaras são valores escolhidos na implementação do programa e o valor inicial de X é tomado com base em alguma preferência do usuário ou pode ser obtida por bibliotecas de aleatoriedade real (tomar os n primeiros bits de um pacote TCP em uma porta predeterminada, por exemplo). Dependendo de qual o tamanho de Z, podemos obter diferentes números, ou um único grande número. Como citado em 3.2.1, para 32 bits temos 624 números.

3.3 Discussão da implementação

3.3.1 Escolha da linguagem

Para decidirmos uma linguagem, precisamos avaliar a diferença entre linguagens interpretadas e linguagens compiladas. Uma linguagem interpretada é, em geral, uma linguagem que a partir de um código escrito por um programador, gera um código intermediário e deste gera um binário. Já uma linguagem compilada, toma o código escrito pelo programador e o compila diretamente. Desta forma, percebe-se que neste intermediário temos a execução de um tempo a mais. Entretanto, como na nossa simulação, não sabemos como o código se comportará (com questão da complexidade), preferimos ter um controle maior do que está sendo compilado. [11]

Desta forma, decidimos utilizar a linguagem C++ e suas bibliotecas padrão, uma vez que estes códigos são constantemente revisados e mantidos por uma comunidade extensa, que nesta, pode se incluir várias empresas (como se pode ver, inclusive, no próprio código fonte das bibliotecas). A biblioteca “random” da linguagem já possui uma implementação própria do Mersenne Twister, e é esta que utilizaremos para a execução do nosso código. Quanto a implementação do Mersenne Twister na biblioteca “random” em si, temos que esta, em propriedades, adapta os valores para os

limites das estruturas de dados primitivas para 32 bits. Assim, especificamente usamos a mt19937. [10]

Não obstante, escolhemos uma linguagem de “baixo nível” pois teremos mais controle do gerenciamento de memória do computador e manipulação de bits, caso necessário.

3.4 Discussão da implementação

O primeiro fato a ser notado é que o computador em si não conhece o número Pi. É funcional na realidade, pois ela própria contém o número pi em sua existência. Porém, como citado na teoria que apresentamos da questão da máquina virtual em 3.1.1, é necessário, para simularmos este aspecto da realidade, conhecermos o número pi de antemão. Portanto, esta é uma simulação que valida as hipóteses e teoria do experimento executado em um mundo real. Não se trata de uma simulação que realmente computa o pi. Avaliando semanticamente, uma simulação pressupõe que o resultado real já está contido dentro dela.

O segundo aspecto da implementação é a existência de estruturas e classes. Uma “struct” em C++ é uma lista de informações que definimos com um nome, e esta é implementada na memória sobre um endereço [12]. Já uma classe possui um princípio similar, porém é possível que implementemos funções específicas para as estruturas. Ou seja, estas funções geralmente são chamadas a partir do nome das estruturas, de tal forma que executem alguma coisa sobre as estruturas. Nesta implementação é possível abstrair a própria Aguha e a coordenada de espaço, por exemplo, em que a Agulha, abstraída como uma classe, recebe do lançamento uma coordenada de espaço. Desta forma, o lançamento em si, que é aplicado na agulha, pode ser representado como uma função que é aplicada sobre a estrutura agulha. Porém, esta é exclusiva para a classe de agulha. Logo, de uma forma simplificada, “toda agulha pode ser jogada”.

O terceiro aspecto é a questão de lançamentos que podem ser feitos dentro do computador. No nosso experimento, estamos nos limitando a 95% de precisão do número pi, o que implica que precisamos fazer lançamentos na ordem dos milhares. Entretanto, é trivial analisar que podemos colocar o computador para fazer o maior número de lançamentos possíveis. Desta forma, o maior número de lançamentos que decidimos colocar (pois existe uma limitação computacional/temporal) foi o número hexadecimal 0xFFFFFFF (16777215 em decimal). Ou seja, 16777215 lançamentos.

O quarto aspecto, é que para demonstrarmos o resultado das computações manipulamos um arquivo do tipo .csv (“comma separated values”) em que podemos analisar os arquivos utilizando nos da “matplotlib” da linguagem Python (uma vez que não é trivial gerar gráficos em C++). Neste sentido, com relação a geração, é preciso notar que em questão de complexidade, estamos deixando o algoritmo mais complexo, uma vez que estamos fazendo a manipulação do arquivo csv.

3.5 Resultados da computação

Fazendo a computação, para 0xFFFFFFF, obtemos um erro de 7.5891e-008. Os resultados do programa estão na figura que apresenta o terminal.

```
O valor de pi e: 3.14159
Erro de norma com relacao ao Pi do Google: 7.5891e-008
```

Figura 6: Temos o resultado apresentado pelo terminal

É possível notar como o valor de Pi converge na figura abaixo.

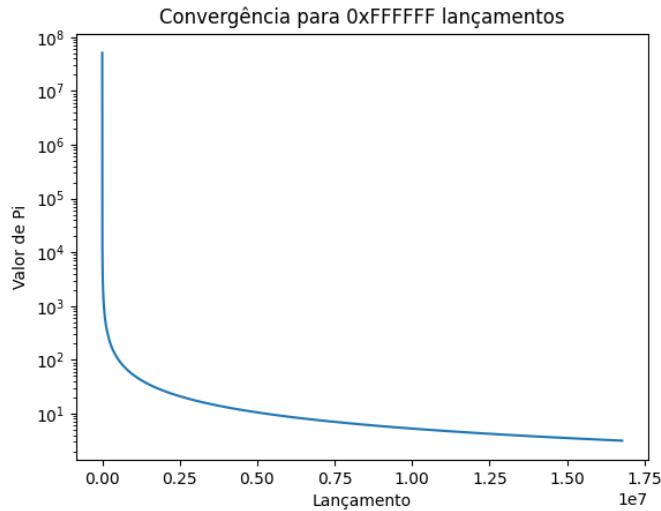


Figura 7: Convergência de pi para cada lançamento

Posteriormente, também podemos avaliar para diferentes lançamentos. Tomando um lançamento inicial de 10, e multiplicando o valor do lançamento por 2 a cada iteração, até 10.000.000 temos o resultado apresentado pela Figura 8. Com isso podemos estimar aproximadamente qual será a convergência do nosso resultado do experimento real utilizando-nos da simulação.

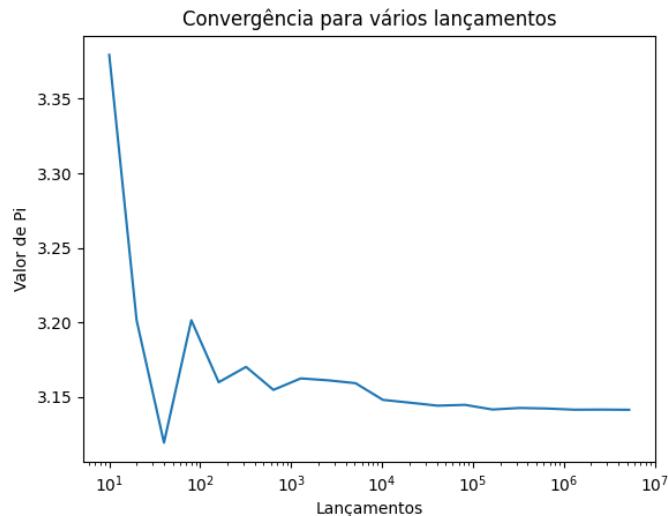


Figura 8: Pi final calculado para cada lançamento

4 Procedimento Experimental

4.1 Descrição do experimento e análises

Para o procedimento experimental para o caso de $L < d$ foram realizadas algumas tarefas. Primeiramente, foram escolhidas medidas tanto para o papel quanto para a agulha que permitissem

uma razão $r = \frac{l}{d} = 0,1966$. Desse modo, para facilitar questões de medida, foi escolhido que a distância entre as linhas presentes no papel A4 seria de $d = 90mm$, exemplo do papel utilizado na imagem abaixo:

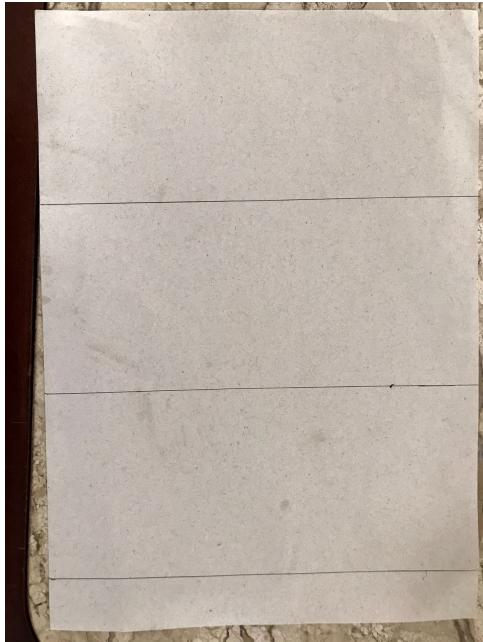


Figura 9: Papel A4 utilizado nos lançamentos

Além disso, para que a razão pudesse ser mantida, o comprimento da agulha obtida foi de $l = 17,7mm$, assim o grupo se utilizou de “palitos de dente” para obter as agulhas na medida necessária, exemplo da agulha utilizada na imagem abaixo:



Figura 10: Agulha utilizada nos lançamentos

Vale notar que tanto para a obtenção das medidas do comprimento e da distância foram feitas a partir de uma régua comum que apresenta $1mm$ como menor valor da escala, logo as incertezas relacionadas aos dois valores serão $\delta l = \delta d = 0,5mm$. Portanto foram realizados os lançamentos até a convergência de um valor aceitável para π medido.

Desse modo, após 4569 lançamentos foi-se obtido $\pi_{medido} = 3,14185$, a incerteza relacionada calculada a partir da equação (2.6) é igual a $\delta\pi = 0,026676$.

O fato do número de lançamentos realizados ter sido maior do que o mínimo esperado, pode estar relacionado a variáveis intrínsecas como o manuseio dos instrumentos por uma pessoa sujeita a variâncias, a superfície de apoio não totalmente plana, sujeiras e consequente alteração no atrito palito-papel, entre outros.

Dentro disso, a imagem abaixo mostra o progresso ao longo dos valores.

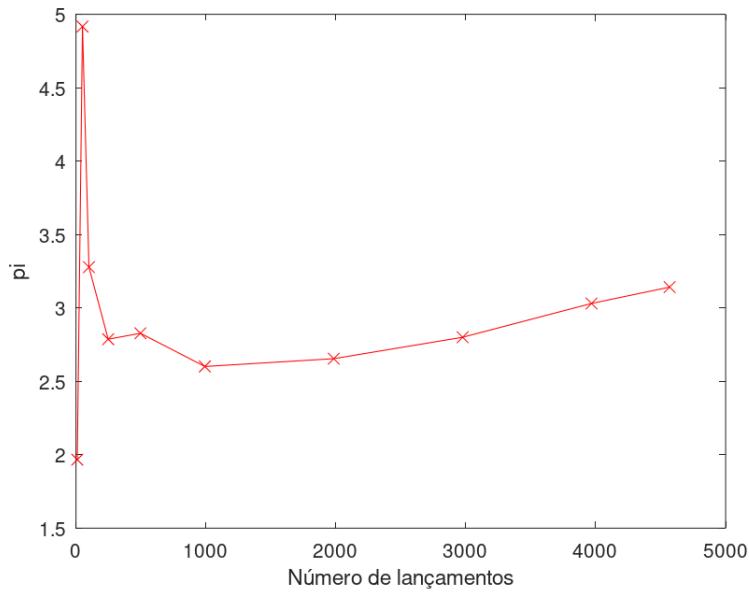


Figura 11: π calculado conforme lançamentos para caso $L < d$

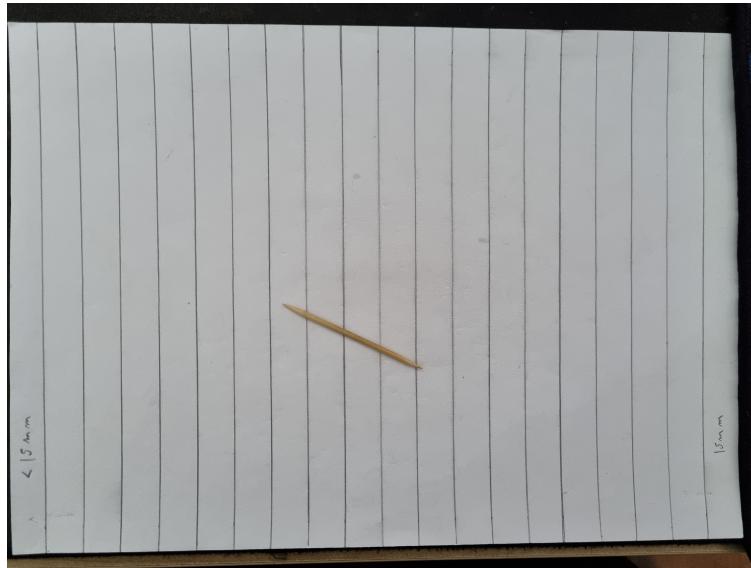


Figura 12: Agulha e papel para o caso $L > d$

Para o procedimento experimental para o caso de $L > d$ foram realizadas algumas tarefas. Primeiramente, foram escolhidas medidas tanto para o papel quanto para a agulha que permitissem uma razão $r = \frac{d}{l} = 0,2336$. Desse modo, para facilitar questões de medida, foi escolhido que a distância entre as linhas presentes no papel A4 seria de $d = 15mm$. Além disso, para que a razão pudesse ser mantida, o comprimento da agulha obtida foi de $l = 64,2mm$, assim o grupo se utilizou novamente de “palitos de dente” para obter as agulhas na medida necessária, buscando reproduzir o contexto do caso anterior apenas com a diferença das medidas adotadas. Como apresentado

anteriormente, os valores de incerteza foram iguais, pois os instrumentos de medição usados foram os mesmos. Desse modo, após 4796 lançamentos foi-se obtido $\pi_{medido} = 3,1399$, com uma incerteza relacionada calculada a partir da equação (2.6) é igual a $\delta\pi = 0,04986$. Esta, novamente, pode ser associada a fatores intrínsecos como superfície e condutor do experimento.

A imagem abaixo mostra o progresso ao longo dos valores.

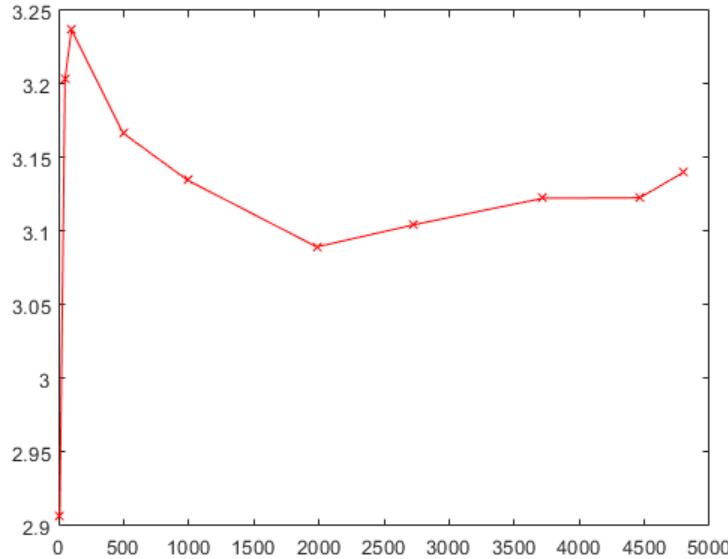


Figura 13: π calculado conforme lançamentos para caso $L > d$

4.2 Discussão do procedimento experimental - análises e incertezas

Durante o processo de lançamento das agulhas, percebe-se uma precisão muito baixa em relação ao valor estimado de π , o que era esperado, como discutido na seção 'Discussão Matemática do Problema'. Já com aproximadamente metade do número de lançamentos necessários, o valor já é razoavelmente próximo da literatura.

Conforme os lançamentos progredem, há pequena variação em torno do valor literário de π , já que a razão entre o número de lançamentos que cruzam a linha e o número de lançamentos que não cruzam a linha tende a permanecer constante. Durante o experimento, é bastante visível esse comportamento, mesmo que aleatório. Foi observado que em certos intervalos de lançamentos há lançamentos seguidos em que há cruzamento da linha no caso $L < d$ e também lançamentos seguidos em que não há cruzamento da linha no caso $L > d$, fenômenos menos frequentes nos dois casos. Também em determinados intervalos, há predominância do caso mais frequente.

Em relação às incertezas da parte experimental, como o procedimento experimental é um processo repetitivo, torna-se difícil realizar o experimento de uma forma totalmente aleatória, isto é, sem viés nos lançamentos da agulha, já que o operador pode tender a lançar a agulha da mesma maneira de forma inconsciente, com movimentos repetidos. Para tentar manter os lançamentos o mais aleatório possível, tanto nos casos para $L < d$ como para os casos $L > d$, os operadores fizeram lançamentos variados, ou seja, visando amenizar o viés. Portanto, para contornar isso, as agulhas foram lançadas variando-se: altura em relação ao papel em que são largadas, ângulo em relação às linhas (rotacionando a agulha antes de largá-la, e também ao soltá-la), inclinação da agulha em relação ao plano do papel (de forma que na grande maioria das vezes a agulha rebatia no papel antes de parar, alterando-se bastante o ângulo a partir do qual o objeto foi solto).

5 Conclusões Gerais

De forma geral, este trabalho apresentou um processo investigativo de um fenômeno matemático que também é um fenômeno físico. Resumidamente, apresentamos as hipóteses matemáticas, testamos-as via simulação computacional e efetuamos o experimento da realidade para extrair um número real de π .

A partir da formulação matemática, é possível averiguar que é possível determinar um experimento que possa mensurar o número π através da análise de vários lançamentos de uma estrutura experimental que possa obedecer aos requisitos. No caso, decidimos fazer a estrutura de tal forma como o experimento foi proposto, porém, seria possível também elaborar algum outro meio probabilístico que fosse análogo ao descrito matematicamente.

Com este mesmo raciocínio, estabelece-se a construção do experimento simulado no computador. Utilizando-se de estruturas de dados e números pseudorrandômicos, é possível avaliar a validade ou não do experimento, servindo inclusive como referência para a execução do próprio experimento. Foi possível avaliar como números randômicos são gerados e a importância de conhecer este processo para não tomar falsas conclusões ou ter expectativas errôneas sobre o processo da simulação e sua aleatoriedade.

Posteriormente, conclui-se por meio dos experimentos reais, que é possível, de fato, extrair o valor de π previamente conhecido a partir deste experimento com uma incerteza suficientemente boa inclusive para ser utilizada em algumas aproximações de engenharia. Por mais que se considere a incerteza, como por exemplo, relacionada a fatores dos lançamentos em si e de quem os conduzem, verifica-se que o experimento é suficientemente robusto a variâncias, tendo sido, portanto, um sucesso, pois conseguimos encontrar um valor para π e observar o fenômeno que se desejava atingir.

Desta forma, através da metodologia proposta neste trabalho, concluímos que é possível através da avaliação das incertezas, avaliação do processo, da metodologia de mensuração das variáveis envolvidas no processo, encontrar o número π com uma precisão razoável através do experimento da Agulha de Buffon, tendo então, portanto, este experimento como sendo uma forma mecânica de aproximarmos do valor real do número π .

6 Anexo A: Códigos

6.1 Cômputo de $N(a)$.

```

1 import numpy as np
2 from matplotlib import pyplot as plt
3
4
5
6
7 #Define entries:
8 epsilon=0.01
9 a=np.linspace(0,1,400) #a\in(0,1)
10 seconda=np.linspace(1,5,400)
11 i=0
12 p=[]
13 secondp=[]
14 N=[]
15 secondN=[]
16 i=0
17 for an in a:
18     p.append((2/np.pi)*np.arccos(an)+((2*an)/(np.pi))*(1-np.sqrt(1-an**2)))
19
20     secondp.append(2/(np.pi*seconda[i]))
21     i+=1
22 i=0
23
24 for pi in p:
25     N.append((1.96**2)*p[i]*(1-p[i])/(epsilon**2))
26
27 secondN.append((1.96**2)*secondp[i]*(1-secondp[i])/(epsilon**2))
28 i+=1
29
30 #Create plot:
31
32
33 fig, axs = plt.subplots(2)
34 fig.suptitle('95% de confiabilidade de erro menor que 0,01.')
35 axs[0].plot(a,N)
36 axs[0].set(ylabel='N(a)')
37 axs[0].grid()
38 axs[0].set_xlim(0,1)
39 axs[0].set_ylim(0,10000)
40
41 axs[1].plot(seconda,secondN)
42 axs[1].plot([4/np.pi, 4/np.pi], [0,9500])
43 axs[1].legend(['N(a)', 'a=4/'+chr(960)])
44 axs[1].set(ylabel='N(a)')
45 axs[1].set(xlabel='a=d/L')
46 axs[1].grid()
47 axs[1].set_xlim(1,5)
48 axs[1].set_ylim(0,10000)
49
50 plt.show()

```

6.2 Código da simulação

```

1 #include <iostream>
2 #include <random>
3 #include <math.h>
4 // #include <stdlib.h>
5 #include <fstream>
6 #include <string>

```

```

7 //na natureza este valor j    existe. precisamos inform -lo para o computador.
8 #define PI 3.14159265359
9
10 struct Space{
11     public:
12         float Theta;
13         float x;
14
15 };
16
17
18 //classe da agulha
19 class Agulha{
20     //Agulha
21     public:
22         float Theta;
23         float x_pos;
24         float L; //tamanho da agulha
25
26     void SetNeedle(float L){
27         this->L = L;
28     }
29
30     bool Throw(float D,Space* Espaco){
31         this->Theta = Espaco->Theta;
32         this->x_pos = Espaco->x;
33         return this->L*cos(this->Theta)/2 > this->x_pos ;
34     }
35 };
36
37
38 // calcula o pi dependendo do caso
39 float Pi_calc(float L, float D, bool Tipo,int n, float p){
40     float pi;
41     if(Tipo){
42         pi = 2*L/(D*p);
43     }
44     else{
45         float r = D/L;
46         pi = (2/p)*(acos(r)+(1/r)*(1-sqrt(1-pow(r,2))));}
47     }
48
49     return pi;
50 }
51
52 float Experimento(float pi, float r, float D, long long Lancamentos){
53
54     std::fstream fout; //ponteiro para o arquivo
55     std::string Nome = "Lancamento_" + std::to_string(Lancamentos)+".csv";
56
57     fout.open(Nome, std::ios::out | std::ios::trunc);
58
59     float L = r*D; //espa o
60     bool Tipo = L < D;
61     int Acertos = 0;
62
63
64     std::random_device rd;
65     std::mt19937 generator{rd()}; //escolha de 32 bits pq      menor
66     std::uniform_real_distribution<double> distribuicaoX(0,D/2);
67     std::uniform_real_distribution<double> distribuicaoT(0,pi/2);
68
69     Space* Espaco = new Space;
70     Agulha* Needle = new Agulha;
71     Needle->SetNeedle(L);

```

```

72     float p =0;
73     float pi_calc = 0;
74     fout << "Lancamento,Pi\n";
75
76     // gera um espaço p/ agulha, joga a agulha, calcula a prob, calcula o pi e joga
77     // no csv
78     for(long long i=0; i<Lancamentos;i++){
79         Espaco->Theta = distribuicaot(generator);
80         Espaco->x = distribuicaox(generator);
81         Acertos += Needle->Throw(D,Espaco);
82         p = (float) Acertos/(float) Lancamentos;
83         pi_calc = Pi_calc(L,D,Tipo,Lancamentos,p);
84         fout << i << "," << pi_calc << "\n";
85     }
86
87     return pi_calc;
88 }
89
90 int main(int argc, char *argv[]){
91     float GooglePi = PI; //pois este é o pi que encontramos ao procurar no google (está na sua calculadora)
92
93     float r = 10; // razão L/D
94     float D = 10; //Distância das bordas
95     long long Lancamento = 0xffffffff;
96     float Pi_max = Experimento(GooglePi,r,D,Lancamento);
97
98     // outros números de lançamentos
99
100    std::fstream fout; //ponteiro para o arquivo
101    fout.open("Tabela_diferentes.csv", std::ios::out | std::ios::trunc);
102    float Pi_teste = 0;
103    fout << "Lancamentos,Pi_obtido\n";
104
105    for(int Lanc=10; Lanc<1e7; Lanc*=2){
106        Pi_teste = Experimento(GooglePi,r,D,Lanc);
107        fout << Lanc << "," << Pi_teste << "\n";
108    }
109
110    std::cout << "O valor de pi é: " << Pi_max << std::endl;
111    std::cout << "Erro de norma com relação ao Pi do Google: " << abs(Pi_max-GooglePi)/GooglePi << std::endl;
112 }
113
114 }
115
116 // pode se ver o repl.it onde o código está publicado aqui: https://replit.com/
117 // @PauloYoshio/AkulhaBuffon?v=1#main.cpp
118 /* recomendo não clicar em run porque a cpu do repl.it não suporta o processo */
119
120 }
```

6.3 Código de análise dos resultados da simulação

```

1 import pandas as pd
2 from matplotlib import pyplot as plt
3 import numpy as np
4
5 Lancgrande = pd.read_csv("Lancamento_16777215.csv")
6 Convergencia = pd.read_csv("Tabela_diferentes.csv")
7
8 n = Lancgrande['Lancamento']
```

```

9 pi = Lancgrande['Pi']
10
11 plt.yscale("log")
12 plt.title("Converg ncia para 0xFFFFFFFF lan amentos")
13 plt.ylabel("Valor de Pi")
14 plt.xlabel("Lan amento")
15 plt.plot(n,pi)
16 plt.show()
17
18
19 n = Convergencia['Lancamentos']
20 pi = Convergencia['Pi_obtido']
21
22 plt.title("Converg ncia para v rios lan amentos")
23 plt.ylabel("Valor de Pi")
24 plt.xlabel("Lan amentos")
25 plt.xscale("log")
26 plt.plot(n,pi)
27 plt.show()

```

7 Referências

- [1] ROSS, Sheldon. **Probabilidade: um Curso Moderno com Aplicações**. 8 ed. Bookmann, 2010.
- [2] Dantas, Carlos A. B. **Probabilidade: um Curso Introdutório**. EDUSP, 2013.
- [3] Physics 116C. <<http://scipp.ucsc.edu/haber/ph116C/NormalApprox.pdf>> Acesso em: 20 de Mar. de 2022.
- [4] Santos, Marcelo M. **Notas de Aula de Análise 2**.
- [5] Mathematical Association of America. <https://www.maa.org/sites/default/files/pdf/upload_library/22/Allendoerfer/1995/Badger.pdf> Acesso em: 22 de Mar. de 2022.
- [6] LAPLACE, Pierre S. **Essai philosophique sur les probabilités**
- [7] KNUTH, David. **Art of Computer Programming, Volume 2: Seminumerical Algorithms** 3 ed. Addison-Wesley Professional, 1997.
- [8] OGATA, K.; **Engenharia de Controle Moderno**. 5 ed. Pearson Prentice Hall.
- [10] C++ Reference - **std::mersenne_twister_engine**. <https://en.cppreference.com/w/cpp/numeric/random/mersenne_twister_engine>
- [11] AHO, et al. **Compilers: Principles, Techniques, and Tools** 2 ed. Addison Wesley Publishing Company, 2006.
- [12] FEOLIOFF, P. **Projeto de Algoritmos**. Disponível em: <<https://www.ime.usp.br/pf/algoritmos/index.html>>